DICIEMBRE 1986

Revista del usuario de Drean Commodore

AÑO 1 Nº 12 ₳ 2,70

REP. ARGENTINA

Llega la Drean Commodore 64C

Port
CONTROL DE LUCES

C-128
¿QUE ES EL CP/M?

Software
JUEGOS Y UTILITARIOS

CLUB DE USUARIOS



NOVIEMBRE 1986 ₳ 3,20 Nº20 REP. ARGENTINA
K64
COMPUTACION PARA TODOS
100 Páginas
K -Test Con Premios
Aplicaciones Científicas
en la Argentina
Software Para PC
ATARI-MSX-CZ-TK
Archivos Comerciales
Para las Commodore
Interfase Universal
Robot Controlado Por una C-64

SUMARIO

NOTAS TECNICAS

REVISION DE SOFT

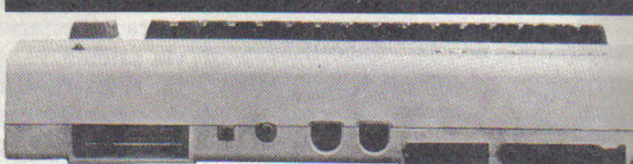
PROGRAMAS

NOTICIAS DREAN COMMODORE

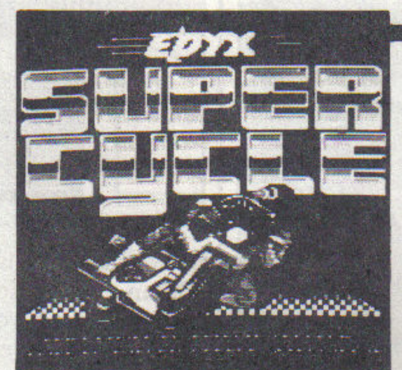
CORREO



Drean ya lo comenzó a fabricar y los usuarios la pueden encontrar en el mercado. Su característica más destacada es la inclusión del Geos en castellano.



Ahí está, detrás de la computadora. Es la conexión con el mundo exterior. Algunos lo saben y lo usan para ampliar la utilidad de la máquina. Otros, la mayoría, ignoran sus posibles aplicaciones.



Continuamos comentando todo el Software que hay en el mercado, desde un utilitario superior hasta la última novedad en juegos.

AÑO 1 Nº 12 NOVIEMBRE DE 1986

Director General
Ernesto del Castillo

Director Editorial
Cristián Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Secretario de Redacción
Ariel Testori

Redacción
Cristián Parodi

Arte y Diagramación
Fernando Amengual
Tamara Migelson

Fotografía
Victor Grubicy
Eduardo Comesaña
Image Bank

Departamento de Avisos
Oscar Devoto
Nelso Capello

Departamento de Publicidad
Guillermo González Aldalur

Drean COMMODORE

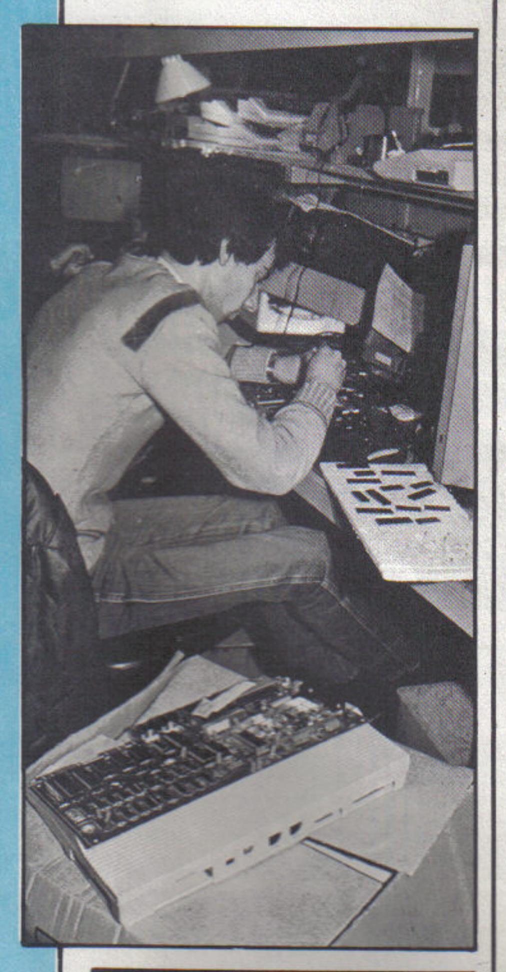
Revista para usuarios de Drean Commodore es una publicación mensual editada por editorial PROEDI S.A., Paraná 720, 5º Pis. (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Reg. Nac. de la Prop. Intelectual E.T., M. Registrada. Precio de este ejemplar, ₳ 2,70. Los ejemplares atrasados se venderán al precio del último número en circulación. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual ISSN0326-8241.

Impresión: Calcotam. **Fotocromo tapa**: Columbia. **Fotocomposición**: Interamericana Gráfica S.R.L.

Distribuido en Capital: Martino, Juan de Garay 358, P.B. Capital. **Distribuidor interior**: DGP, Hipólito Yrigoyen 1450, Capital Federal. T.E.: 38-9266/9800.

PRODUCCION ANUAL DE DREAN COMMODORE

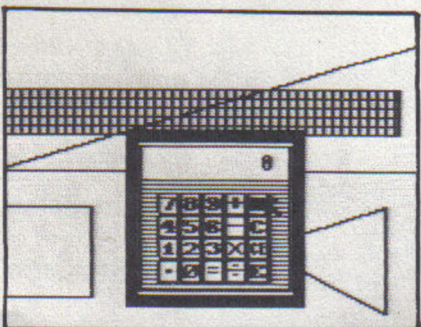
Drean , a través de su Gerente de Publicidad y Promoción, Felipe Mc Gough, informó, entre otras cosas, la perspectiva de producción de Computadoras Drean Commodore para el año 1987
Mc Gough precisó que se estiman en unas 100.000 unidades a partir de marzo de 1987.
Además, se refirió a la salida de la Amiga y de la PC de Commodore.
La primera fue considerada por Mc Gough como "una máquina más grande que una Home Computer pero que tiene la potencia de una PC". La otra será "ciento por ciento compatible con la PC de IBM, y tendrá un precio que dará mucho que hablar", agregó.



MODEM PARA LA 64 Y 128

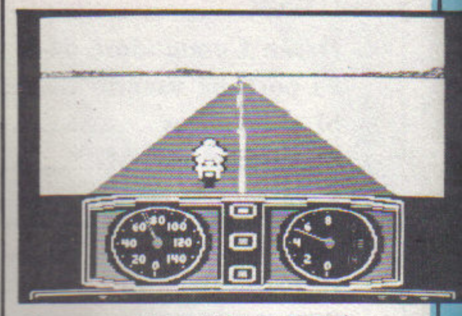
Siscotel S.A., representante de Delphi, una de las bases de datos más grandes del mundo, distribuye los modems TAIHAHO compatibles para la Drean Commodore 64 y la C-128 (trabajando en modo 64) junto con el software de comunicación respectivo.
De esta manera cualquier usuario de los equipos Drean Commodore tendrá acceso (vía cable telefónico) a la base de datos que Siscotel tiene en nuestro país.
Los servicios que se le brindan al usuario van desde la posibilidad de tomar todo tipo de información (noticias, bibliotecas, comerciales, etc.) hasta conectarse con Delphi USA vía satélite.
En el próximo número haremos una descripción más a fondo de este importantísimo servicio.

GEOS EN CASTELLANO



Drean lanzará junto con la nueva Drean Commodores 64C el sistema operativo GEOS, del cual hemos hablado en números anteriores.
Este GEOS es exactamente igual que el americano pero tiene una importantísima ventaja: fue desarrollado íntegramente en castellano.
Como recordarán GEOS son las siglas de Graphic Environment Operating System. Este sistema es muy fácil de operar ya que podemos seleccionar la función a través de gráficos predefinidos.

MOTOCICLISMO



A través de DATAGAMES SOFTWARE nos hemos enterado de este excelente juego de competición, que a más de uno le provocará vértigo. El juego se llama "Super Cycle" y es extraordinario. Otro de los juegos acercados fue "Landing on TB19", "Parallax", "Color Me", e "Indoor Sport". Todos ellos podrán encontrarlos en nuestra Revisión de Software de este número. También nos llegó "World Game".

dBASE II

CHIPS COMPUTACION nos ha entregado el dBASE II para la Commodore 128 en modo CP/M.
Como verán en la Revisión de Software de este número, el dBASE es una de las bases de datos más potentes para computadoras con microprocesadores de 8 bits.
Permite trabajar con hasta 65535 registros con 32 campos por cada uno de ellos.

CLUB DE DREAN COMMODORE EN DELPHI

El Club de usuarios de Drean Commodore estará a partir del 1° de diciembre suministrando sus servicios bajo el rubro "clubes y grupos" de Delphi.
Esta es una base de datos nacional e internacional que brinda todo tipo de servicios a los usuarios de computadoras.
Sólo basta tener un modem telefónico para poder ingresar a esta base.

NOTICIAS DREAN COMMODORE

Gracias a esta nueva tecnología todos los usuarios de equipos Drean Commodore podrán tomar todo tipo de información del Club a través de Delphi.

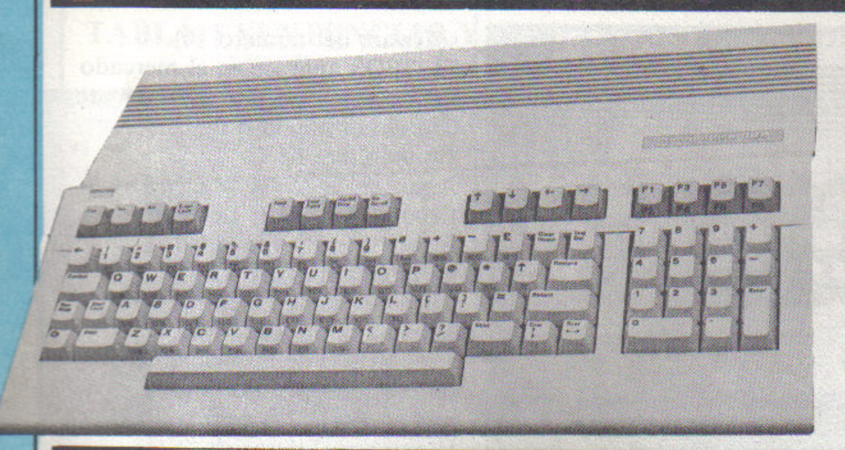
DREAN COMMODORE 128 Y DRIVE 1571

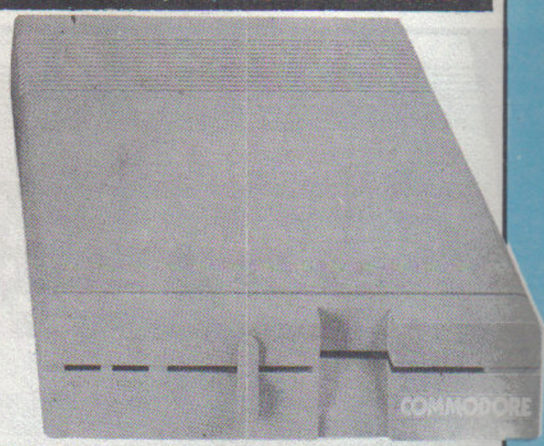
Otro de los lanzamientos previstos por Drean es el de la Drean Commodore 128 y el drive 1571. Estarán en el mercado nacional a partir de fin de año.

La ventaja de la Drean Commodore 128 residirá en la posibilidad de trabajar en tres modos (modo 64,128 y modo CP/M).

En el primero la máquina se comporta exactamente igual a una Drean Commodore 64.

En el segundo suministra su nuevo Basic 7.0 y la posibilidad de trabajar con 128 kb de memoria RAM. En tanto, en el tercer modo, podemos utilizarla como una PC trabajando bajo el control del sistema operativo CP/M (que explicamos en este número).





DREAN COMMODORE 64 C

Drean ya la comenzó a fabricar y los usuarios la pueden encontrar en el mercado. Su característica más destacada es la inclusión del GEOS en castellano.



La primera característica sobresaliente del equipo reside en la posibilidad de trabajar con el nuevo sistema operativo GEOS. GEOS es las sigla de Graphic Environment Operating System. Aquí cada función está representada por un "grafiquito" al cual se lo denomina Icono. Como nota al margen podemos decir que los Iconos fueron tablas de madera utilizadas en Rusia y en Grecia en donde se representaba a la divinidad.

Lo fundamental del GEOS es que el usuario puede efectuar cualquier tipo de operación con sólo seleccionar, con el joystick, el ícono deseado.

No debemos omitir que la Drean Commodore 64 puede, de igual manera, trabajar con este sistema operativo (ver revisión de software del número 10).

El GEOS aparece en el mercado norteamericano simultáneamente con el lanzamiento de la Commodore 64 C.

Aquí, en la Argentina, la Drean Commodore 64 C tiene el formato, muy similar al de la Commodore 128.

Internamente es idéntica a la Drean Commodore 64, disponiendo de la misma capacidad de memoria y utilizando el mismo Basic 2,0 desarrollado por la empresa Microsoft.

Otra diferencia visible, además del diseño similar al de la 128, reside en su teclado. Es mucho más suave al tacto que el de la 64 y Drean Commodore 16.

Con respecto a los periféricos, es posible conectar toda la gama disponible para la 64 en la 64 C. Así, por ejemplo, podemos conectar la disquetera 1541 o la 1571, al Datassette, el mouse, impresoras serie Commodore (MPS 801, MPS 803 o la nueva MPS 1000) o el modem telefónico.

A través de éste último podemos, vía cable telefónico, conectarnos a bases gigantes de datos, las cuales están funcionando en vuestro país, y de las que hablaremos en el próximo número.

En otro orden de cosas la 64 C puede utilizar el nuevo monitor TV de Drean, logrando así una configuración completa.

Este nuevo televisor puede ser utilizado como un televisor color normal o como un monitor profesional.

Junto con la computadora se presenta el correspondiente manual del usuario (mucho más ampliado), una guía

introductoria y un cassette de prueba.

Para el neófito la guía introductoria puede resultar de gran utilidad. En ella se le explica cómo realizar el conexionado y cómo dar sus primeros pasos utilizando la Drean Commodore 64 C. Por último cabe destacar el respaldo técnico que dispone el usuario de la Drean Commodore 64 y 64 C. Dicho respaldo consiste en service técnico y la posibilidad de seleccionar un programa de entre más de 4000 títulos distintos que actualmente existen para la 64.

EL BASIC DE LA DREAN COMMODORE 64 C

La tabla 1 representa los comandos disponibles de la nueva Drean Commodore 64.

Como hemos mencionado anteriormente, el respaldo técnico otorgado por Drean es, podríamos decir, revolucionario.

Si la Drean Commodore 64 C tiene algún problema, Drean la cambia por una nueva.

TABLA 1 SENTENCIAS Y COMANDOS DE LA DREAN COMMODORE 64 C

Palabra	Abreviatura.	
ABS	A SHIFT B	Determina el valor absoluto de un número.
AND	A SHIFT N	Realiza la función lógica AND.
ASC	A SHIFT S	Devuelve el código ASCII del argumento.
ATN	A SHIFT T	Ejecuta la función arcotangente.
CHR $	C SHIFT H	Devuelve el caracter asociado al argumento.
CLOSE	CL SHIFT O	Cierra un archivo.
CLR	C SHIFT L	Resetea todas las variables definidas hata el momento.
CMD	C SHIFT M	Envia a otro periférico la información de salida.
CONT	C SHIFT O	Continúa con la ejecución del programa interrumpido.
COS	NINGUNA	Realiza la función matemática coseno.
DATA	D SHIFT A	Almacena temporalmente información.
DEF	D SHIFT E	Define una función.
DIM	D SHIFT I	Guarda espacio en memoria para los arreglos (vectores y/o matrices).
END	E SHIFT N	Culmina la ejecución de un programa.
EXP	E SHIFT X	Realiza la función exponencial.
FN	NINGUNA	Trabaja junto con la función DEF.
FOR	E SUJET O	Ejecuta una serie de instrucción hasta que la variable índice llegue a un determinado valor.
FRE	F SHIFT R	Indica la cantidad de memoria libre
GET	G SHIFT E	Toma un caracter desde el teclado.
GET#	NINGUNA	Toma un caracter desde el archivo actualmente abierto a través de la sentencia OPEN.
GOSUB	GO SHIFT S	Efectúa el salto a una subrutina.
GOTO	G SHIFT O	Transfiere el control del programa incondicionalmente.
IF	NINGUNA	Evalúa un predicado y ejecuta ciertos comandos si éste es verdadero.
INPUT	NINGUNA	Permite el ingreso desde el teclado.
INPUT #	I SHIFT N	Permite el ingreso desde un archivo lógico.
INT.	NINGUNA	Devuelve la parte entera de un número.
LEFT$	LE SHIFT F	Devuelve el string más a la izquierda de una serie de caracteres.
LEN	NINGUNA	Determina la longitud de una variable alfanumérica.
LET	L SHIFT E	Permite la asignación de una variable (esta sentencia es opcional).
LIST	L SHIFT I	Lista el programa almacenado en memoria.
LOAD	L SHIFT O	Carga un programa en la memoria.
LOG	NINGUNA	Realiza la función matemática logaritmo.
MID$	M SHIFT I	Devuelve el string ubicado en determinada posición dentro de un string mayor.
NEW	NINGUNA	Borra el programa almacenado en la memoria.
NEXT	N SHIFT E	Trabaja junto con la sentencia FOR. Incrementa la variable índice en lo que determine el paso puesto en STEP.
NOT	N SHIFT O	Realiza la función matemática NOT.
ON	NINGUNA	Trabaja junto con las sentencias GOSUB y GOTO. Transfiere el control a una determinada línea acorde al valor de una variable.

Palabra	Abreviatura	
OPEN	O SHIFT P	Abre un archivo lógico.
OR	NINGUNA	Realiza la función matemática OR.
PEEK	P SHIFT E	Lee el valor almacenado en una deteminada dirección de memoria.
POKE	P SHIFT O	Pone un determinado valor en una dirección de memoria
POS	NINGUNA	Posiciona el cursor en cualquier columna de la línea actual.
PRINT	?	Muestra contenidos de variables o leyendas.
PRINT #	P SHIFT R	Pone información en el archivo actualmente abierto.
READ	R SHIFT E	Lee los datos almacenados en sentencias DATA.
REM	NINGUNA	No interviene en la ejecución del programa. Ayuda a la documentación del mismo.
RESTO-RE	RE SHIFT S	Actualiza los datos almacenados en sentencias DATA.
RETUR-N	RE SHIFT T	Devuelve el control al programa principal.
RIGTH$	R SHIFT I	Devuelve el string ubicado más a la derecha de un string principal.
RND	R SHIFT N	Ejecuta la función random.
RUN	R SHIFT U	Ejecuta un programa.
SAVE	S SHIFT A	Graba un programa en disco o casette.
SGN	S SHIFT G	Devuelve el signo de una variable.
SIN	S SHIFT I	Realiza la función matemática seno.
SPC	S SHIFT P	Posiciona el cursor en una determinada columna de la línea en curso.
SQR	S SHIFT Q	Realiza la operación matemática raiz cuadrada.
STATUS	ST	Indica el estado de los periféricos.
STEP	ST SHIFT E	Trabaja junto con la sentencia FOR-NEXT. Indica el incremento de la variable índice.
STOP	S SHIFT T	Para la ejecución del programa.
STR$	ST SHIFT R	Convierte un número en una expresión alfanumérica.
SYS	S SHIFT Y	Ejecuta un programa escrito en código máquina.
TAB	T SHIFT A	Posiciona el cursor.
TAN	ÑINGUNA	Realiza la función tangente.
THEN	T SHIFT H	Trabaja jutno al IF. Permite que se ejecuten las sentencias si la condición es cierta.
TIME	TI	Devuelve el valor del reloj de tiempo real en unidades de 1/60 segundos.
TIME$	TI$	Devuelve el valor del reloj interno en formato horas, minutos y segundos.
USR	U SHIFT R	Pasa valores a una rutina en Assembler.
VAL	V SHIFT A	Convierte un número tipo alfanumérico a numérico.
VERIFY	V SHIFT E	Verifica si ha sido almacenado correctamente el programa en disco o en cinta.
WAIT	W SHIFT A	Suspende la ejecución del programa hasta que se cumplan ciertas condiciones.

EL PORT DEL USUARIO

Ahí está, detrás de la computadora. Es la conexión con el mundo exterior. Algunos lo saben y lo usan para ampliar la utilidad de la máquina. Otros, la mayoría, ignoran sus posibles aplicaciones.

En algunas ocasiones sólo la utilizamos para conectar el famoso "reset", esa cajita mágica con un pulsador que nos permite abandonar un programa que no es posible brekear.

Muchos son los que desconocen las posibles aplicaciones del port del usuario y lo que es posible hacer a través de él.

Básicamente el port del usuario es un dispositivo que le permite conectarse con el mundo exterior, es decir todo lo que está afuera de la computadora.

A través del Port podemos conectar nuestra 64 ó 128 con un modem telefónico, una impresora tipo centronics, un sintetizador de voz o, inclusive, conectarla a otra computadora.

Para ello hace falta saber cómo debemos enviar información hacia el Port, con lo cual necesitamos conocer los registros involucrados con el Port.

Estos —los registros— forman parte de las CIAs e indican, entre otras cosas, quién enviará datos y quién los recibirá.

Para comprender mejor lo que estamos diciendo, observen el gráfico de la figura 1, que representa a una de las CIA. Estas son las que se encargan de comunicarse con la disquetera o impresora y, además, son las que nos permiten escribir nuestros programas a través del teclado. Una de ellas (la CIA número 2) nos permite, entre otras cosas, enlazarnos con el mundo exterior via el Port del usuario.

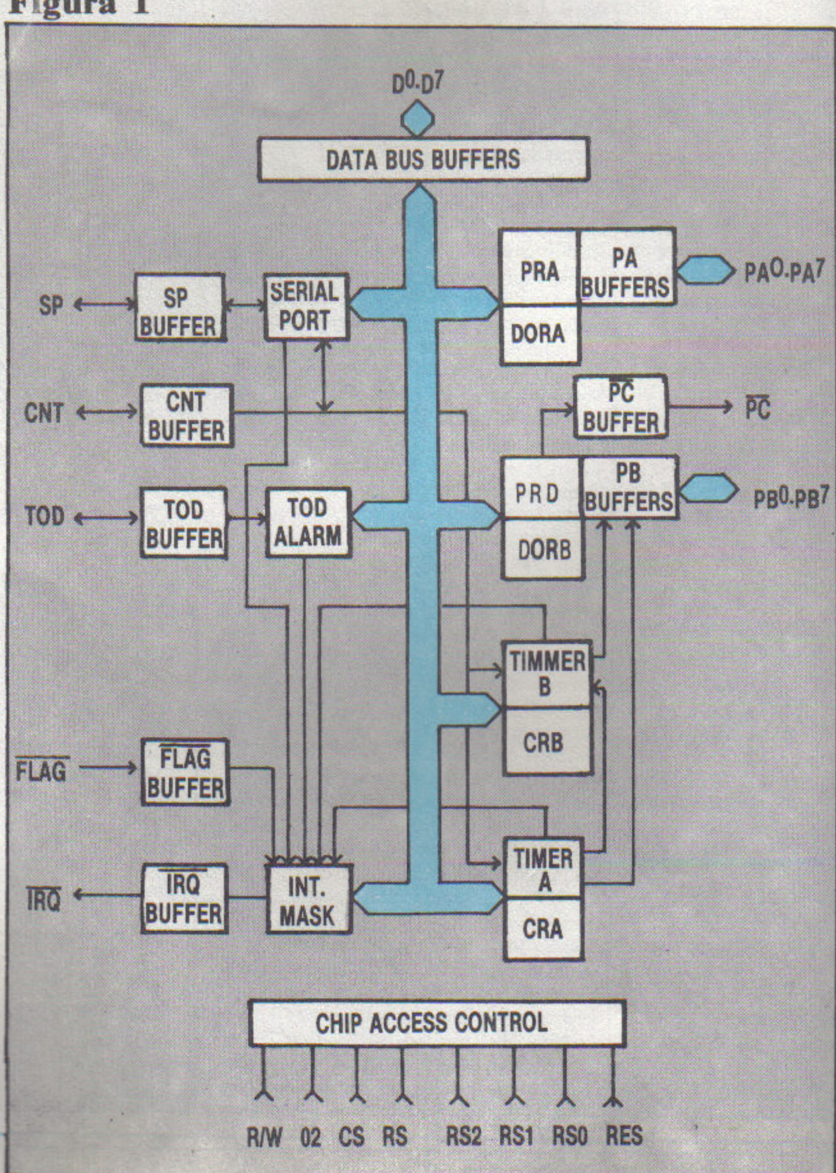
Como verán en dicho gráfico, se representa una serie de órganos de control. Cada uno de ellos tiene una función determinada. Nosotros sólo nos ocuparemos del DDRB y del PRB de la CIA 2, que nos permitirá realizar nuestro gran proyecto.

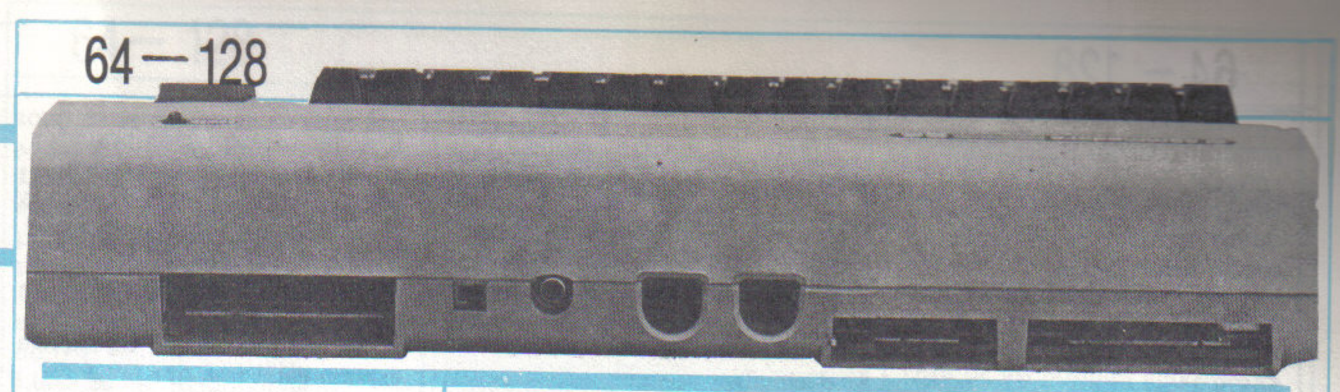
Debido al diseño de la CIA, sólo es posible enviar hasta 8 bits de datos o recibir la misma cantidad.

Es decir que podemos enviar o recibir solamente un byte de información o realizar una especie de mezcla entre entradas y salidas.

Si lo necesitamos, podemos "dividir" el port de tal manera que, por ejemplo, cuatro bits sean utilizados como salidas y que los restantes cuatro bits sean

Figura 1





utilizados como entradas.
Si observan la figura 2 y 3 notarán que indicar que un bit puede ser salida o entrada significa decir cómo se relacionará la computadora con el mundo exterior.
Cada bit indica si una de las ocho líneas serán usadas como salida o como entrada.

MANO Y CONTRAMANO

Como hemos mencionado, dentro de la CIA existe un órgano (en realidad son dos y la explicación es idéntica para cada uno) que se encarga de indicar qué líneas se utilizarán como entrada y cuáles como salida.
A este órgano se lo identificó con un nombre que representase su función. Por ello se lo bautizó con Data Direction Register (DDR-registro de direcciones).
Tanto en la 64 como en la 128, a este registro se lo puede encontrar en la dirección hexadecimal $DD03 (56579). El contenido de ésta le indica a la CIA 2 qué líneas serán usadas como salida y cuáles como entrada.
Esto se realiza a través de cada uno de los bits de la dirección. Si está en "1" indica salida, mientras que, con un "0", entrada.
En la figura 4 se representa la correspondencia entre los bits de esta dirección y las líneas de datos.
Si ponemos un "1" en este registro a través de:
POKE 56579,1
pondremos la línea 0 de entrada
Si efectuamos:
POKE 56579,255
todas las líneas serán usadas como salida. En contraposición:
POKE 56579,0

Figura 2

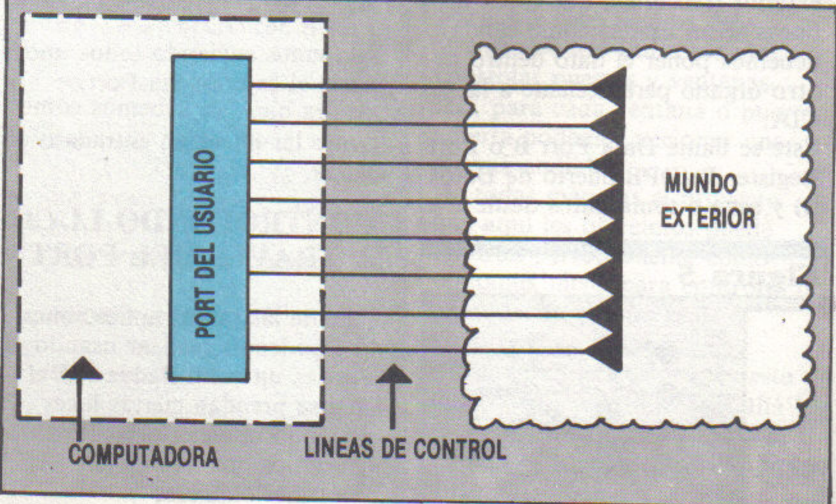


Figura 3

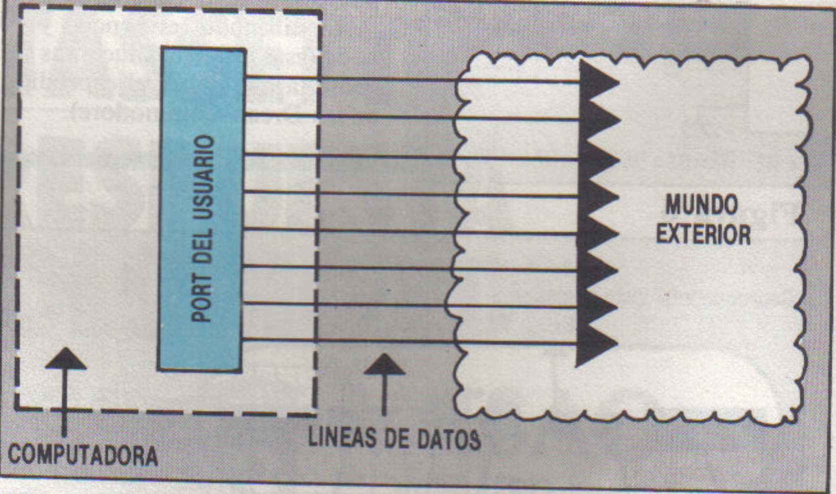
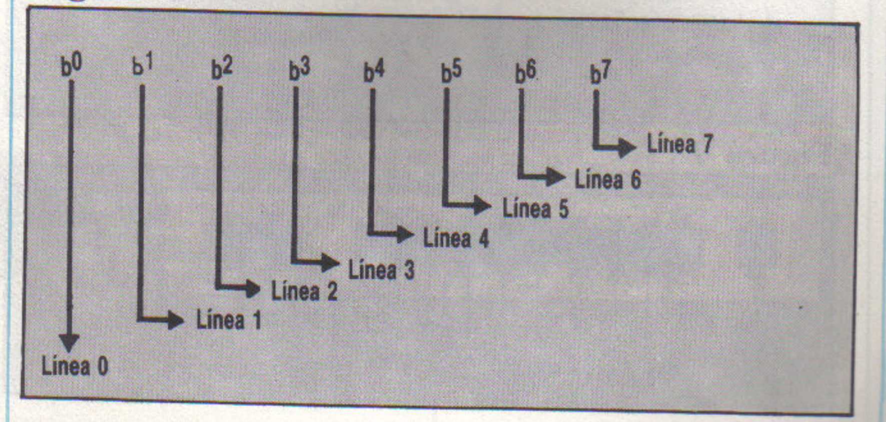


Figura 4



pondrá todas las líneas como entrada.
Luego de establecer la dirección de los datos, podemos enviar o recibir información.
Nosotros, antes de continuar, determinaremos todas las líneas como salidas a través de POKE 56579,255.
Para enviar información hacia el exterior (recuerden que todas las líneas serán usadas como salida) debemos poner el dato dentro de otro órgano perteneciente a la CIA.
Este se llama Data Port B o Port Register B (DPB-puerto de Datos B) y saca o toma datos desde afuera. En nuestro caso él enviará información hacia el exterior, como consecuencia de haber preparado todas las líneas como salida.
Ese dato debemos ponerlo en la dirección asociada a este registro, la cual es $DD01 (56577 decimal).
Si ponemos un valor de 255 a través de:
POKE 56577,255
estaremos enviando todos unos hacia el exterior via Port.
Ahora bien, ya sabemos cómo poner las líneas en entrada o en salida, ¿y...?

CONTROLANDO LUCES A TRAVES DEL PORT

Una de las tantas aplicaciones que podemos realizar usando el Port es un controlador con el cual se prendan ciertas luces tomando una cierta secuencia (como los que hay en los "boliches bailables").
Nosotros desarrollaremos el proyecto a nivel experimental, es decir utilizando resistencias y Leds (esas pequeñas lucesitas como la que indica el encendido en las Drean Commodore).

Figura 5

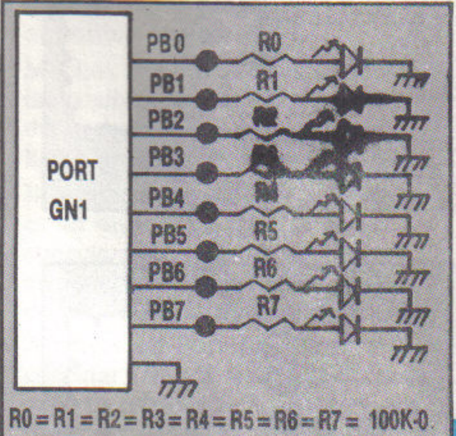


Figura 6

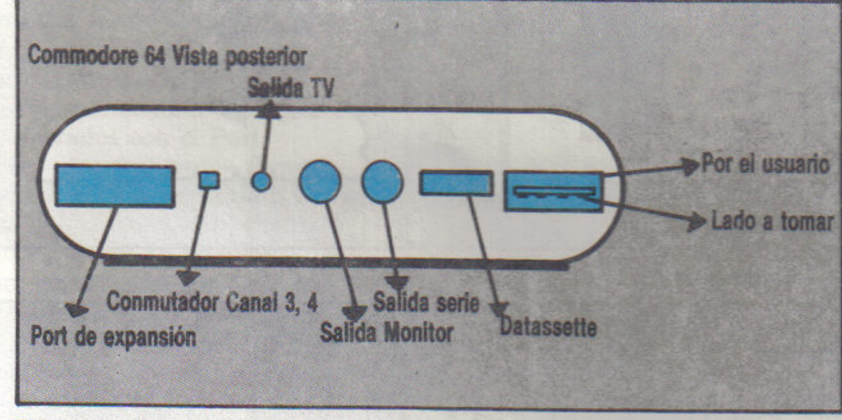


Figura 7

Vista posterior del port

A

PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7

Realizar la adaptación posterior es cosa fácil. Sólo necesitamos insertar entre las luces de color y la computadora una unidad que adapte las diferencias de potencia entre ambas. Los elementos que necesitamos son pocos: un protoboard o plaqueta experimental (esa que tiene muchos agujeritos para poner alambrecitos, resistencias, etcétera), alambre fino para realizar las conexiones, un conector simple hembra de doce contactos (el que conectaremos en el Port), ocho resistencias de 100 k Ohms y ocho leds de distintos colores.
El circuito completo corresponde a la figura 5. Las letras PB0, PB1, PB2,..., PB7 representan, ni más ni menos, a las líneas de datos mencionadas anteriormente.
Ustedes deberán, antes que nada, localizar estas líneas en la Port. Por ello tomen el manual del usuario, estírense y vean cómo es la Port. Habrán notado que la plaqueta es doble faz (doble lado), tiene 12 contactos de un lado y doce del otro.
Utilizaremos algunos de los doce del lado de abajo, viendo a la Commodore desde atrás.
La figura 6 indica qué lado se debe tomar. En ese lado sólo utilizaremos 8 patas empezando desde la tercera y culminando en la décima segunda.
La figura 7 indica qué patas utilizaremos (las que representan PB0, PB1,...,PB7). La pata denominada GND es tierra, la cual servirá para que se cierre el circuito.
Una vez armado el circuito, debemos hacer el programa que prenda determinadas luces.
El listado 1 es una primera versión de controlador de luces. Este las va prendiendo en forma secuencial una por una.
Repite la secuencia 10 veces, las prende y apaga 3 veces y vuelve a repetir la secuencia.
La explicación del programa es la que sigue a continuación:
Línea 10: le decimos a la Port que todas las líneas serán usadas de salida (computadora-luces).
Línea 20: La variable T indica el

total de veces que se repetirá la secuencia (10 veces)
Línea 30: La variable 1 selecciona una de las ocho líneas por donde se enviará un "1" que prenderá un led.
Línea 40: A través del POKE, enviamos el "1" a la PRB, es decir al mundo exterior.
Línea 50-60: Retardamos la ejecución del programa para que se pueda ver cómo se cambia de led a led.
Línea 70: Enviamos otro "1" a otra línea.
Línea 80: Iniciamos la segunda secuencia.
Línea 90: Prendemos todas las luces poniendo 255 en el PRB.
Línea 110-120: Retardo.
Línea 130: Finalizamos la ejecución del programa.
Si modifican el valor final de la sentencia FOR de la línea 50, verán cómo se prenden más rápido las luces.
Prueben ustedes mismos modificar las secuencias para

Listado 1

```
10 POKE 56579,255
20 FOR T=1 TO 10
30 FOR I=0 TO 7
40 POKE 56577,2↑I
50 FOR D=1 TO 100
60 NEXT D
70 NEXT I: NEXT T
80 FOR T=1 TO 3
90 POKE 56577,255
100 FOR D=1 TO 100
110 NEXT D
120 NEXT T
130 STOP
```

producir distintos efectos.
Una de ellas podría ser que las luces se vengan prendiendo desde los extremos hacia el medio de una por vez.
O que prendan al azar durante un tiempo y que luego comiencen con otra secuencia.
Otra de las posibles aplicaciones puede ser la de control, tipo alarma.
Para este caso la Port del usuario podría utilizarse para controlar puertas y ventanas.
Así, para cada ventana o puerta abierta podemos accionar una alarma audible.
De todas maneras el proyecto que aquí les ofrecimos puede servirles perfectamente como puntapie inicial para controles de procesos en tiempo real.
En próximos números continuaremos explorando esta misteriosa "puerta al más allá", y les explicaremos cómo hacer nuestro propio reset a bajo costo y super fiable.

SUPER DIRECTORIO

directorio.12	prg	6	19	12	2049
PROGRAMAS.12	seq	30	4	19	
PORT.12	seq	6	0	34	
HAL.12	seq	31	1	5	
ahorcado	prg	2	8	26	2049
TRUCOS.12	seq	6	14	14	
IF-THEN.12	seq	6	6	27	

Comp.: Drean Commodore 16 y 64
Tipo: Utilitario
Conf.: Disk Drive y/o impresora

A través de este programa podrán conocer el directorio del disco junto con informaciones varias.
Estas corresponden al tipo de programa (SEQ, REL, etc.), la cantidad de bloques que ocupa, la ubicación en disco (es decir el track y sector en donde se lo puede localizar) y la dirección de inicio del programa.
Además nos indica de qué espacio libre dispone el disco.
SUPER DIRECTORIO también permite que enviemos la información hacia la impresora, donde se imprimirá a simple o doble ancho. En caso de no tener impresora, podemos verlo en la pantalla. Al comienzo, se nos preguntará si deseamos ver el directorio sobre la pantalla o en impresora.
Cada opción se selecciona a través de las letras I o P.
Para el caso de salida por impresora (I), se nos preguntará si deseamos imprimirlo a simple o doble ancho. Como en el caso anterior esto se determina con las letras 'S' o 'D'.
Luego se nos pedirá que oprimamos cualquier tecla cuando el disco esté en el drive.
La última pregunta será respecto a la dirección inicial de los programas. Podemos ver la de todos, de algunos o de ningún programa almacenado en disco.
Finalmente comenzarán a listarse todos los datos de cada programa (track y sector en donde se encuentran en disco, tipo de programa, tamaño y dirección inicial).
La dirección inicial es muy importante para saber el número que acompaña a la instrucción SYS.
En muchos de los juegos sucede que éstos deben ser cargados con la opción "1" del LOAD y luego ejecutar un SYS junto con la dirección especificada.
Si ponemos otro valor, no podremos correr el programa. A través de SUPER DIRECTORIO visualizaremos rápidamente la dirección de inicio.

VARIABLE UTILIZADA:

Nombres	Descripción
ND$	Variable de control de impresión.
FT$	Vector: indica tipo de programa.
F$	Matriz: almacena los programas en disco.
A$	Uso auxiliar
I	Variable índice
LB$,HB$	Indican parte baja y alta de la dirección inicial.
NM	Variable índice
DV	Indica tipo de dispositivo.
EZ,EZ$,TR,SE	Indican tipo de error y track en sector en donde ocurrió.

DESCRIPCION DEL PROGRAMA

Línea	Descripción
10-60	Inicialización de variables y tablas.
70-160	Selecciona salida por impresora o por pantalla.
170-240	Inicializa y toma información del disco.
250-700	Módulo principal. Toma datos y los almacena en las tablas.
710-930	Imprime directorio del disco.
940-1010	Repite opciones.
1020-1-50	Subrutina: Verifica que no haya errores.

PEQUEÑO GLOSARIO

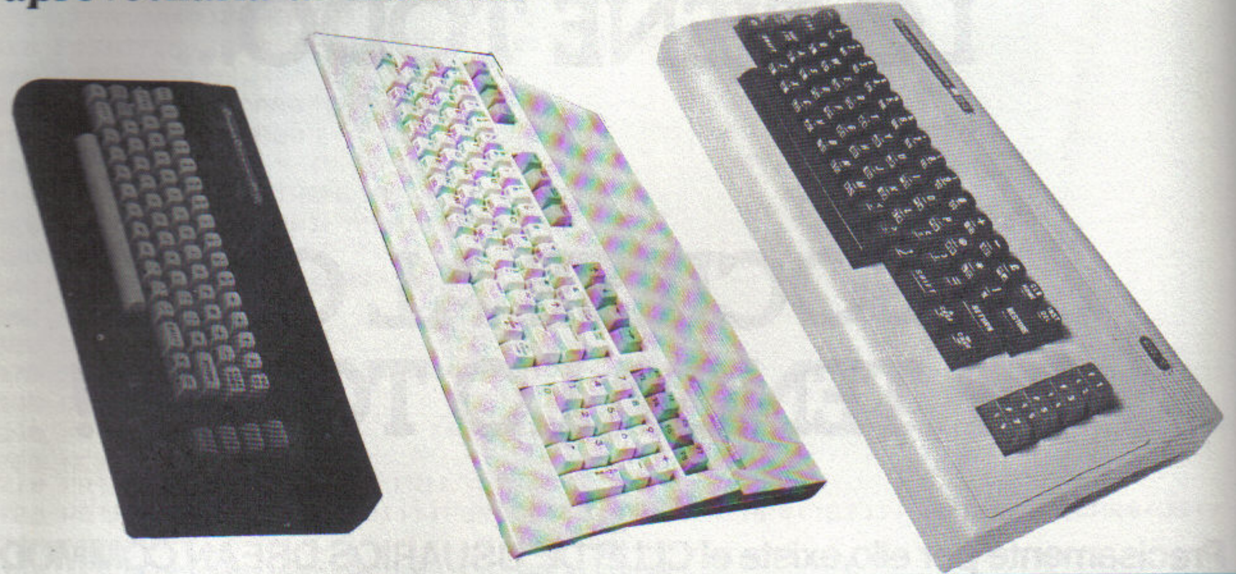
SEQ: Archivo de organización secuencial.
REL: Archivo de organización relativa o comúnmente llamada al azar.
TRACK: Circunferencia concéntrica ubicada en el disco.
SECTOR: Sección de 256 bytes. Varias de estas secciones constituyen un Track.
SYS: Comando que ejecuta un programa escrito en código máquina.
Dirección de inicio: Dirección que le indica al intérprete Basic a partir de qué dirección se deberá cargar el programa.

```
10 ND$=""
20 REM ND$=CHR$(8)
30 DIM FT$(5):FORI=0TO5:READA$:FT$(I)=A$:NEXT
40 DATA DEL,SEQ,PRG,USR,REL,DEL
50 DIM F$(144,5)
60 FE=664
70 PRINTCHR$(147);CHR$(30);CHR$(17);"SALIDA EN PANTALLA O IMPRESORA (P/I) ?"
80 GETA$:IFA$=""THEN80
90 DV=3:IFA$="I"THENDV=4
100 IFA$="P"THEN140
110 PRINTCHR$(17)"SIMPLE O DOBLE ANCHO (S/D) ?"
120 GETA$:IFA$=""THEN120
130 BG$=CHR$(15):IFA$="D"THENBG$=CHR$(14)
140 PRINTCHR$(17)"COLOQUE EL DISCO Y LUEGO OPRIMA ALGUNA  TECLA":PRINT
150 GETA$:IFA$=""THEN150
160 PRINTCHR$(145)"UN MOMENTO POR FAVOR...
170 OPEN15,8,15:PRINT#15,"I":GOSUB1020
180 OPEN8,8,8,"$0,S,R"
190 GOSUB1020
200 FORI=1TO142:GET#8,A$:NEXT
210 FORI=143TO160:GET#8,A$:N$=N$+A$:NEXT
220 FORI=161TO162:GET#8,A$:ID$=ID$+A$:NEXT
230 GET#8,A$:FORI=164TO165:GET#8,A$:OS$=OS$+A$:NEXT
240 FORI=166TO254:GET#8,A$:NEXT
250 CT=8
260 NM=NM+1
270 IFCT=8THENCT=1:GOTO300
280 CT=CT+1:GET#8,A$,A$:FL=ST
290 IFFL<>0THEN480
300 GET#8,A$:IFA$=""THENA$=CHR$(133)
310 FL=ST:IFFL<>0THEN480
320 TY$=FT$((ASC(A$)AND191)-128)
330 GET#8,A$:IFA$=""THENA$=CHR$(0)
340 TR$=RIGHT$("  "+STR$(ASC(A$)),2)
350 GET#8,A$:IFA$=""THENA$=CHR$(0)
360 SC$=RIGHT$("  "+STR$(ASC(A$)),2)
370 FL$="":FORI=3TO18:GET#8,A$:FL$=FL$+A$:NEXT
380 FORI=19TO27:GET#8,A$:NEXT
390 GET#8,LB$,HB$
400 BL=ASC(LB$+CHR$(0))+256*ASC(HB$+CHR$(0))
410 IFTY$<>"DEL"THENFE=FE-BL
420 BL$=RIGHT$("    "+STR$(BL),3)
430 IFTR$=" 0"THEN480
440 F$(NM,0)=FL$:F$(NM,1)=TY$:F$(NM,2)=TR$:F$(NM,3)=SC$:F$(NM,4)=BL$
450 F$(NM,5)="     "
```

```
460 IFTY$="PRG"THENF$(NM,5)="-----"
470 GOTO260
480 CLOSE8
490 GOSUB1020
500 IFF$(NM,0)=""THENNM=NM-1:GOTO500
510 FE$=RIGHT$("    "+STR$(FE),3)
520 PRINTCHR$(145);"DIRECCION INICIAL:      ":PRINTCHR$(17);" 1) DE TODOS"
530 PRINT" 2) DE ALGUNOS":PRINT" 3) NINGUNO"
540 GETA$:IFVAL(A$)=0THEN540
550 IFVAL(A$)>2THEN770
560 PRINT"♥":IFA$="1"THENPRINT"UN MOMENTO"
570 FORI=1TONM
580 IFF$(I,1)<>"PRG"THEN700
590 IFA$="1"THEN630
600 PRINTF$(I,0);" (S/N)"
610 GETA$:IFA$=""THEN610
620 IFA$<>"S"THENPRINT"□";:GOTO700
630 SA$=F$(I,0)
640 OPEN8,8,8,"0:"+SA$+",P,R"
650 GOSUB1020
660 GET#8,LB$,HB$
670 SA=ASC(LB$+CHR$(0))+256*ASC(HB$+CHR$(0))
680 CLOSE8
690 F$(I,5)=RIGHT$("       "+STR$(SA),5)
700 NEXT
710 PRINT"■■■ ESPERE UN MINUTO"
720 PRINTCHR$(28);SPC(21);:FORQQ=1TO14:PRINTCHR$(163);:NEXT:PRINTCHR$(30);
730 PRINT:PRINT"Y";
740 IFDV=4THENPRINT" ACTIVE LA IMPRESORA";
750 PRINT" OPRIMA UNA TECLA"
760 GETA$:IFA$=""THEN760
770 IFDV<>4THENPRINTCHR$(147)
780 IFDV=4THENIFND$=""THENOPEN6,4,6:PRINT#6,CHR$(21):CLOSE6
790 OPEN4,DV
800 PRINT#4,BG$;"┌─────────────────────┬───┬───┬───────────┐";ND$
810 PRINT#4,BG$;"|";N$;"|";ID$;"|";OS$;"|FR SEC:  ";FE$;"|";ND$
820 PRINT#4,BG$;"├─────────────────────┴─┬─┴─┬─┴──┬──────────┤";ND$
830 PRINT#4,BG$;"|NOMBRE   :            |TIP |TR |SC |BLK |START|";ND$
840 PRINT#4,BG$;"├──────────────────────┼────┼───┼───┼────┼─────┤"
850 FORI=1TONM
860 FL$=F$(I,0):TY$=F$(I,1):TR$=F$(I,2):SC$=F$(I,3):BL$=F$(I,4)
870 IFTY$="DEL"THEN900
880 PRINT#4,BG$;"|";FL$;"|";TY$;"|";TR$;"|";SC$;"|";BL$;"|";F$(I,5);"|";ND$
890 PRINT#4,BG$;"|                      |    |   |   |    |     |";ND$
900 NEXT
910 PRINT#4,BG$;"└──────────────────────┴────┴───┴───┴────┴─────┘";ND$
920 PRINT#4:CLOSE4
930 CLOSE15
940 IFDV=4THENPRINT"IMPRIME NUEVAMENTE (S/N) ?"
950 IFDV=3THENPRINT"OTRO DIRECTORIO (S/N) ?"
960 GETA$:IFA$="S"THEN770
970 IFA$<>"N"THEN960
980 PRINTCHR$(145)"NUEVO DIRECTORIO (S/N) ?"
990 GETA$:IFA$="S"THENRUN
1000 IFA$="N"THENPRINT"♥":END
1010 GOTO990
1020 INPUT#15,EZ,EZ$,TR,SE:IFEZ=0THENRETURN
1030 T$=CHR$(157)+CHR$(32)
1040 PRINTCHR$(18) EZ;T$;EZ;EZ$;TR;T$;SE
1050 CLOSE8:CLOSE15
READY.
```

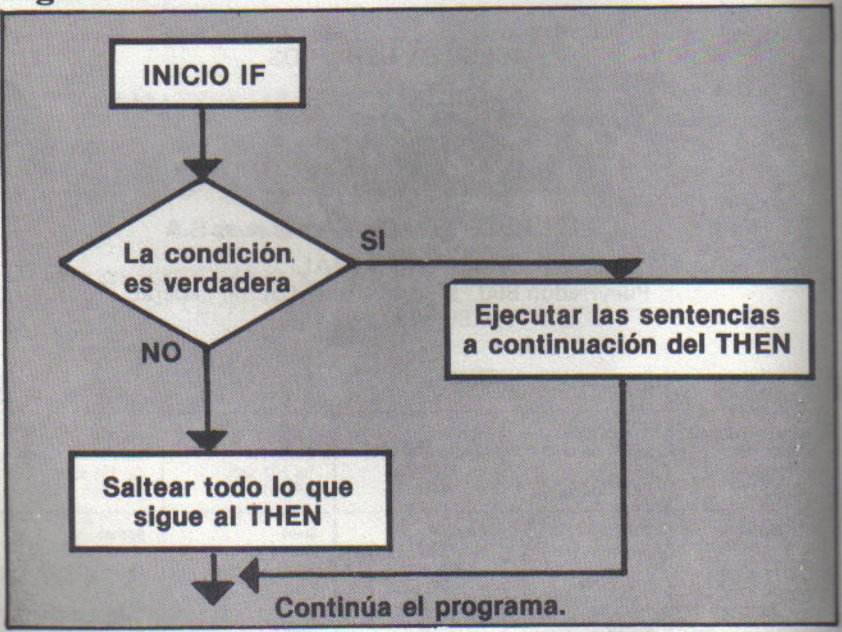
SI... ENTONCES

Este comando es una de las sentencias más importantes de todo lenguaje. Explicamos a fondo cómo trabaja y cómo aprovecharla al máximo.



Si tuviéramos que traducir esta sentencia diríamos que se trata, más o menos, de lo siguiente: Si (IF) tal cosa es cierta, entonces (THEN) hacer esto.
Gracias a esta sentencia podemos detectar condiciones que, por algún motivo, necesitamos saber si son ciertas.
Nosotros tenemos la capacidad de determinar cuándo una cosa que nos interesa se cumple o no. Si decimos, por ejemplo, "Si llueve vamos al cine" sabemos que iremos a ver una película si algunas gotas caen sobre nosotros.
Pero, la computadora, necesita "alguien" que le indique cuándo debe hacer un proceso o ejecutar una serie de sentencias. Al igual que la condición "si llueve", el intérprete necesita una condición similar.
Cuando se esté ejecutando la sentencia IF-THEN, el Basic determina si la condición es cierta. Si ello ocurre se ejecutan todas las sentencias que están después del THEN.
Si, en cambio, la condición es falsa, se saltean todas las sentencias que están después del THEN y se ejecutan las que están de ahí en más.
La figura 1 representa el funcionamiento interno del IF-THEN. Como ven, sólo se ejecutan las sentencias a continuación del THEN cuando se verifica que la condición es cierta.
En caso contrario salteamos a esas sentencias y continuamos la ejecución del programa.

Figura 1



Condiciones y operadores

Las condiciones que hemos mencionado pueden ser de distinto tipo. Así, por ejemplo, podemos chequear si dos variables son iguales.
De esta manera saldría una condición del tipo:
variable1 = variable2
con lo que el IF-THEN quedaría:
IF variable1 = variable2 THEN...
Esto quiere decir que si la viariable1 es exactamente igual a la variable2 (nuestra condición), entonces se deberán ejecutar las sentencias que siguen a continuación del THEN.
Con las variables string (las alfanuméricas) también es posible realizar comparaciones.
Como explicamos en el número anterior, estas comparaciones se llevan a cabo tomando los códigos ASCII del caracter (para mayor información consulten la nota "Cómo ordenar alfabéticamente" del número anterior).

Figura 2

X	Y	XORY
F	F	F
F	V	V
V	F	V
V	V	V

Figura 3

X	Y	XAND Y
F	F	F
F	V	F
V	F	F
V	V	V

Figura 4

X	NOT X
F	V
V	F

Otras de las sentencias que se involucran con el IF-THEN son AND y OR. A través de éstas podemos formar una condición que esté integrada por varias subcondiciones.
Siguiendo en nuestro ejemplo "Si llueve voy al cine" podemos también decir que "Si llueve o el cielo está nublado voy al cine". Es decir que si una de las dos subcondiciones es cierta (si llueve o si está nublado) nosotros iremos al cine.
Para este tipo de condiciones se utiliza la sentencia OR. Este es un operador lógico que se rige a través de la "tabla de verdad" representada en la figura 2.
Noten que la función OR es verdadera cuando alguno de los operandos es verdadero.
Por ejemplo en:
IF A = 1 OR B = 1 THEN STOP
finalizaremos la ejecución del programa a través de la sentencia STOP cuando alguna de las dos variables sean uno (noten que si

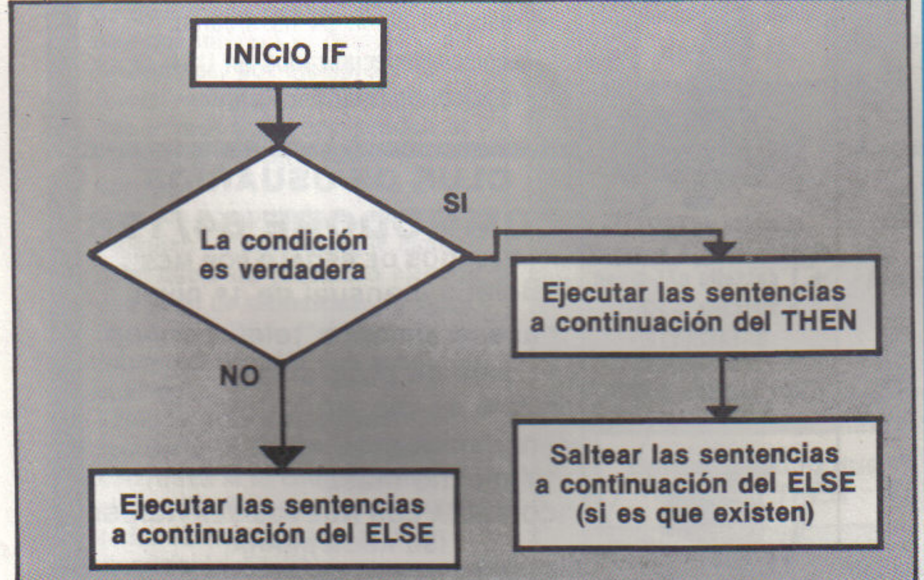
ambas son iguales a uno la condición también es cierta).
En este caso el intérprete Basic determina si la condición A = 1 OR B = 1 es verdadera, actuando como lo representamos en la figura 1.
Ahora bien, qué hubiese sucedido si, en cambio, nuestro ejemplo si hubiera sido "vamos al cine si llueve y hace frío".
Para este caso utilizamos la función lógica AND. La figura 3 representa a su tabla de verdad.
A diferencia del OR, esta función será verdadera únicamente cuando los operandos sean ambos verdaderos.
Como en nuesto ejemplo, iremos al cine cuando esté nublado y además haga frío.
En Basic podemos hacer por ejemplo:
IF A = 1 AND B = 0 THEN...
Si A = 1 y si B = 0 (es decir si la condición es cierta) entonces se ejecutarán las sentencias que continúan luego del THEN.
No debemos olvidar que tanto el OR como la AND también son utilizados para realizar operaciones aritméticas.
Es decir que podemos hacer, por ejemplo, asignaciones (darle a una variable un cierto valor) del tipo:
A = B AND 64
C = (A OR B) AND 128

Este tema lo abordaremos en próximas notas.
Las condiciones también pueden utilizar ambas funciones lógicas.
Por ejemplo:
IF (A = 1 AND B > 0) OR (E < 1 AND O = 1) THEN...
trabajará de la siguiente manera:
1) El intérprete Basic evalúa la subcondición A = 1 AND B > 0.
Si ésta es cierta ejecuta lo que está a continuación del THEN ya que de por medio existe un OR (recuerden que con el OR basta con que alguna de las dos sea verdadera).
2) Si la primera subcondición es falsa, se evalúa la segunda subcondición (E < 1 AND O = 1).
Si ésta es verdadera ocurre como en el caso anterior.
3) Si ambas son falsas se saltean las sentencias a continuación del THEN, tal como se representa en la figura 1.
También es posible mezclar condiciones y variables. Por ejemplo, es posible comprobar si una variable alfanumérica es mayor que otra y si una númerica es mayor que cero; es decir:
IF EMPLEADO$ > A$ AND NUMERO > 0 THEN COSTO = COSTO * 100.
En este caso sólo efectuaremos la operación COSTO = COSTO * 100 si la variable EMPLEADO$ es mayor que A$ y que, además, se cumple que la variable NUMERO es mayor que cero.
Por último disponemos de la función lógica NOT, representada a través de la figura 4. Esta función es la negación: si algo es verdadero la pone en falso y viceversa.

ELSE

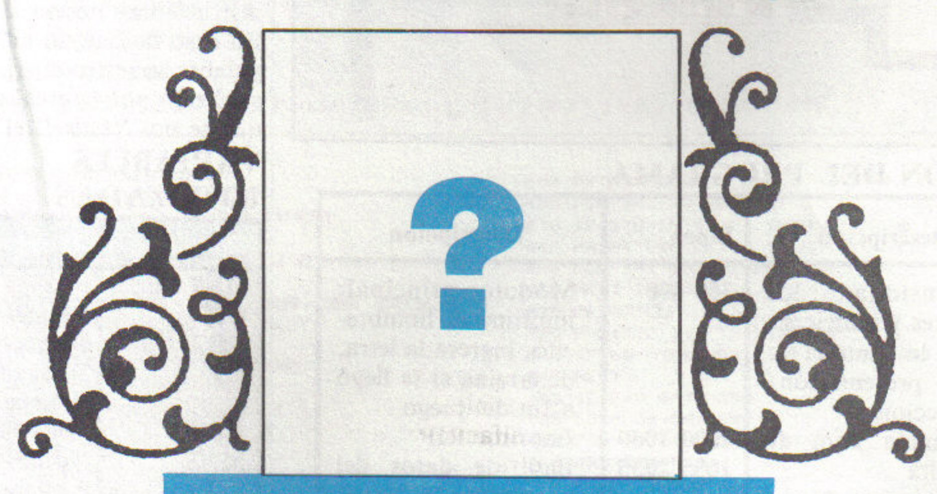
Algunos Basic, como el de la Drean Commodore 16 y Commodore 128 disponen del IF-THEN-ELSE.
Su formato es IF condición THEN (sentencias que se ejecutan si es verdadera) ELSE (sentencias que se ejecutan si es falsa).
Esta sentencia se diferencia de la anterior ya que tiene la posibilidad de ejecutar sentencias por verdadero y por falso.
Como lo indicamos en la figura 5, primero se evalúa la condición. Si ésta es verdadera ejecutamos lo que sigue al THEN deteniéndonos, cuando encontramos el ELSE si éste existe.
Si, en cambio, la condición es falsa, saltea todo lo que está a continuación del THEN y ejecutamos, si éste existe, a partir del ELSE.
La otra ventaja que surge de trabajar con el ELSE es que los programas pueden fácilmente adoptar una estructura clara.
Rápidamente sabemos qué es lo que hace el programa si la condición es verdadera o qué cosa hace si ésta es falsa.
El ELSE de la Drean Commodore 16 solamente puede ejecutar un grupo reducido de sentencias, limitadas por la máxima cantidad de caracteres que es posible poner por línea.
La Commodore 128, por el contrario, dispone de dos sentencias especiales que le indican al intérprete desde dónde hasta dónde se deben ejecutar las sentencias si la condición es verdadera o si es falsa.
En próximos números trataremos este tema desde el punto de vista matemático. Es decir cómo realizar operación A AND B, NOT A AND C, etcétera.

Figura 5



BUSCADO

Drean Commodore busca al usuario del año '86



CAUSA

Premiaremos a las aplicaciones más originales de las Drean Commodore 16 y 64 en cualquier ámbito o actividad. Por ejemplo, se tendrán en cuenta el uso inteligente de estas máquinas en el campo profesional, educativo, comercial, científico y doméstico.

Quienes deseen participar deberán enviar a nuestra redacción, Paraná 720, 5to. Piso, Cap. Federal (1017), una descripción del uso que se le dio al equipo (si es posible con fotos).

El cierre del certamen será el 28/11/86

RECOMPENSA

1er. PREMIO: UNA COMPUTADORA DREAN COMMODORE 64 C

2do. PREMIO: UNA DISKETTERA DREAN COMMODORE 1541

AHORCADO



Comp.: Drean Commodore 64
Conf.: Básica
Tipo: Juego

En este juego nosotros deberemos adivinar la palabra que selecciona la computadora. Esta puede elegirse de entre 4 categorías distintas, a saber: computadoras, deportes, coches, naciones y temas populares. Carguen y ejecuten el software, donde encontrarán las instrucciones necesarias.
En caso de que no adivinemos la palabra, nuestro cuerpo irá subiendo por la escalera hasta que se nos ''estire'' el cuello.

DESCRIPCIÓN DEL PROGRAMA

Línea	Descripción	Línea	Descripción
5	Dimensiona los vectores y matrices	180 890	Módulo principal: imprime el hombrecito, ingresa la letra, determina si se llegó a fin del juego
10	Borra la pantalla	1000 1050	Imprime RIP
26 150	Pone presentación/ instrucciones	1055 2035	Base de datos del programa.
155 160	Determina item de consulta		
165 175	Pide número de opción		

VARIABLES UTILIZADAS

Nombre	Descripción
HN$,PR$, C$	Vectores
TD$	Matriz
C$	Variable auxiliar: lee el teclado.
I,J	Variable índice
MN$	Nombre del participante
C1	Variable auxiliar
TT	Copia del tiempo de juego.

```
5 DIM HM$(8),TD$(6,4),PR$(17),C$(25)
10 PRINT"█"
26 PRINT"■":POKE53280,1:POKE53281,1
28 PRINT"██████"TAB(13)"*************"
29 PRINTTAB(13)"*"TAB(25)"*"
30 PRINTTAB(13)"* ♠|┌_─♠┬   *"
31 PRINTTAB(13)"*"TAB(25)"*"
32 PRINTTAB(13)"*"TAB(25)"*"
33 PRINTTAB(13)"*************"
34 POKE 53272,23:PRINT"████████████ ⁄ECESITA INSTRUCCIONES  ?";
35 GET C$:IF C$="" THEN 35
37 IF C$<>"S" THEN 110
40 PRINT"██┬N ESTE JUEGO SE DEBERA DESCUBRIR LA PA LABRA ELEGIDA POR LA COM";
45 PRINT"PUTADORA. ¬STA   LA SELECCIONARA DE UN BANCO DE DATOS   QUE SE";
50 PRINT" ENCUENTRA EN LA MEMORIA.        "
55 PRINT"█ OMO EN EL JUEGO DEL AHORCADO, POR CADA LETRA NO ADIVINADA NUESTRA";
60 PRINT" CABEZA SE IRA ACERCANDO A LA SOGA, LA CUAL ESTA LISTA PARA ESTI";
65 PRINT"RARNOS.                   "
70 PRINT"██ |UENA SUERTE !!!"
100 PRINT"   ████ |IPEE CUALQUIER TECLA      ████"
105 GETC$:IF C$="" THEN 105
110 PRINT"████████,NGRESE SU NOMBRE        ?";:GOSUB350:NM$=LEFT$(C$,9)
120 FOR I=1 TO 8:READ HM$(I):NEXT
125 FOR I=1 TO 6: FOR J=1 TO 4
130 READ TD$(I,J):NEXT:NEXT
135 A$="██████":B$="██████████████"
150 POKE53272,23:POKE59468,14:PRINT"██┬LIGE EL TEMA A TRATAR :██"
155 CT$(1)="COMPUTADORAS":CT$(2)="DEPORTES":CT$(3)="COCHES":CT$(4)="NACIONES"
```

```
160 CT$(5)="POPULAR    "
165 FOR A=1 TO 5:PRINTA;"█. ";CT$(A):NEXT
170 PRINT"██     |U ELECCION, "+NM$+" ";:GOSUB950:C=VAL(C$)
171 IF C>=1 AND C<6 THEN 175
172 PRINT"¬NGRESE LA CATEGORIA DESEADA,"+NM$:FORZ=1TO399:NEXT:GOTO 150
175 PRINT " █████     ,N MOMENTO POR FAVOR.....":TT=TI
185 FOR I=0 TO (C-1)*20+INT(RND(1)*20)
190 READ WD$:NEXT
200 IF TI-TT<120 THEN 200
205 LM$="":PR$="":LG$=""
210 FORI=1 TO LEN(WD$)
215 IF MID$(WD$,I,1)=" " THEN PR$(I)=" ":PR$=PR$+" ":GOTO 225
220 PR$(I)="-":PR$=PR$+" "
225 NEXT:POKE53272,21
227 MN$=LEFT$(NM$,1)
228 IF ASC(MN$)>95 THENNM$=CHR$(ASC(MN$)-128)+MID$(NM$,2)
```

```
230 VP=17:HP=0:M=0:PRINT"█";
235 FOR I=1 TO 16:PRINT"█";:NEXT
240 FOR I=1TO8:PRINT:PRINTHM$(I);:NEXT
245 FOR I=1 TO 7
250 PRINT"█ □█───── □█";:NEXT:PRINT"█";
255 PRINT"█ □█  █───█ █"
260 PRINT"█";TAB(34);"█▃  ████ | ███⌒███⌣□□□█▃ ";
265 FOR I=1 TO 23:PRINT"██ ██";:NEXT
270 PRINT"██ █";:POKE 33767,160
275 PRINT"█PALABRA     : ";CT$(C);"□
280 FOR I=1 TO 8 :PRINT"█" :NEXT
285 PRINTTAB(18);"█DESCONOCIDAS  "
290 PRINTTAB(18);"──────────────"
295 PRINT"██████";PR$
300 FOR I=1 TO 17:PRINT"█";:NEXT
305 PRINTTAB(16);LM$
310 PRINT"████████LETRA                    ████
    ████████";:GOSUB950:G$=C$+" "
315 IF MID$(G$,2,1)=" " GOTO 350
320 PRINT"□□□□"
325 FOR I=1 TO 20
330 IF I/2=INT(I/2)THEN PRINT"█";
335 PRINT"SOLO UNA LETRA, ";NM$;"□"
340 FOR J=1 TO 50:NEXT
345 NEXT I:GOTO 310
350 G$=LEFT$(G$,1)
355 IF G$>="A" AND G$<="Z" GOTO 385
360 PRINT"□□□□":FOR I=1 TO 20
365 IF I/2=INT(I/2)THEN PRINT"█";
370 PRINT"ESTO NO ES UNA LETRA!!□"
375 FOR J=1 TO 50:NEXT
380 NEXT I:GOTO 310
385 FOR I=1 TO LEN(LG$)
390 IF G$=MID$(LG$,I,1) GOTO 400
395 NEXT I:GOTO 430
400 PRINT"□□□□"
405 FOR I=1 TO 20
410 IF I/2=INT(I/2) THEN PRINT"█";
415 PRINT"ESA LETRA YA HA SIDO ELEGIDA!!□"
420 FOR J=1 TO 50:NEXT
425 NEXT I:GOTO 310
430 PRINT"□□                         "
435 F=0:FOR I=1 TO LEN(WD$)
440 IF G$=MID$(WD$,I,1) THEN PR$(I)=G$:F=1
445 NEXT
450 IF F=0 GOTO 650
455 LG$=LG$+G$:PR$=""
460 FOR I=1 TO LEN(WD$)
465 PR$=PR$+PR$(I):NEXT
470 PRINT"██████";PR$
475 PRINT"                    □"
480 FOR I=1 TO 20
485 IF I/2=INT(I/2) THEN PRINT"█";
490 PRINT"BIEN   !!!"+NM$+"!□"
495 FOR J=1 TO 50:NEXT:NEXT
500 IF PR$=WD$ GOTO 565
505 PRINT"████████ADIVINAS LA PALABRA (S/N)?   ██";
510 GOSUB950
512 IF LEFT$(C$,1)<>"S" THEN 310
515 PRINT"████████PALABRA               "
520 GOSUB 950:G$=LEFT$(C$,LEN(WD$))
525 IF G$=WD$ GOTO 565
530 PRINT"████████                       □"
535 FOR I=1 TO 20
540 IF I/2=INT(I/2) THEN PRINT"█";
545 PRINT"OLEEE...,"+NM$+"□"
550 FOR J=1 TO 50:NEXT:NEXT
555 PRINT"██████████              "
560 GOTO 310
565 PRINT"██████████████████████████████";
```

```
570 FOR I=1 TO 10
575 PRINT"♦ ♦ ♦ ♦ ♦ ♦ "+B$;
580 PRINT"            ♦"+B$;
585 PRINT"♦ TU GANAS   "+B$;
590 PRINT"            ♦"+B$;
600 PRINT"♦ ♦ ♦ ♦ ♦ ♦ "+B$;
605 PRINT"□□□□□";
610 PRINT" ♦ ♦ ♦ ♦ ♦ ♦"+B$;
615 PRINT"♦            "+B$;
620 PRINT"  ■TU GANAS■ ♦"+B$;
625 PRINT"♦            "+B$;
630 PRINT" ♦ ♦ ♦ ♦ ♦ ♦"+B$;
635 PRINT"□□□□□";:NEXT
640 PRINT:PRINT"■■■■■■OTRO JUEGO"+NM$+"?"
645 GOTO 925
650 PRINT"■■■■■■■                          □"
655 FOR I=1 TO 20
660 IF I/2=INT(I/2)THEN PRINT"■";
665 PRINT"OLEEE......."+NM$+"!□"
670 FOR J=1 TO 50:NEXT:NEXT
675 LG$=LG$+G$:LM$=LM$+G$
680 PRINT"■":FOR I=1 TO 10:PRINT"■":NEXT
685 PRINTTAB(18)+LM$:M=M+1
690 IF M=9 GOTO 800
695 FOR I=1 TO 2:VP=VP-1:PRINT"■";
700 FOR J=2 TO VP:PRINT"■";:NEXT
705 IF VP=1 THEN PRINT"□";
710 PRINTSPC(HP+1)+"■";
715 FOR J=1 TO 8
720 PRINT"■"+HM$(J)+"■■■■";
725 NEXT J:PRINT"■     ";:NEXT I
730 FOR I=1 TO 4:HP=HP+1:PRINT"■";
735 FOR J=2 TO VP:PRINT"■";:NEXT
740 IF VP=1 THENPRINT"□";
745 PRINTSPC(HP)+"■";
750 FOR J=1 TO 8
755 PRINT"■ "+HM$(J)+"■■■■■";:NEXT:NEXT
760 IF M<8 GOTO 310
765 PRINT"■";SPC(HP);
770 FOR I=1 TO 8
775 PRINT"■  "+HM$(I)+"■■■■■■";:NEXT
780 GOTO 310
800 PRINT"■■■■■■■■■■";SPC(34);
805 FOR I=1 TO 6:FOR J=1 TO 4
810 PRINTTD$(I,J)+A$;:NEXT
815 PRINT"□□□□";:NEXT
820 PRINT"■■■";SPC(34);
825 FOR I=1 TO 16:PRINT"■■  ▌  ";
830 FOR J=3 TO 8
835 PRINT A$+HM$(J);:NEXT
840 PRINT"□□□□□■■■■";:NEXT
850 FOR I=1 TO 1000:NEXT
860 PRINT"■□◤   ◥"+A$;
865 PRINT"■ R  |"+A$;
870 PRINT"■ I+|"+A$;
875 PRINT"■P||"+A$;
880 PRINT"■  |■ "+A$;
885 PRINT" ■  ■ "+A$;
890 PRINT" ■  ■ "+A$;
895 PRINT" ■  ■ ";
900 FOR I=1 TO 3000:NEXT
905 PRINT"■■■■■■■LA PALABRA SECRETA ERA:"
910 PRINT"■■■■■■■■■■■■■■"+WD$
915 PRINT"■■■■MEJOR SUERTE LA PROXIMA VEZ,
   ";NM$;"!!!"
920 PRINT"■■■■OTRO JUEGO ? ";
925 RESTORE
930 FOR I=1 TO 32:READ WD$:NEXT
935 GET C$:IF C$="" THEN 935
937 IF C$="S" THEN 150
940 PRINT"■■■GRACIAS POR EL JUEGO !!!!":FOR Z=
   1TO3500:NEXT:RESTORE:END
950 C$="":REM  SUBRUTINA INGRESO
952 PRINT"■■";:C$=""
955 GET C1$:IF C1$="" THEN 955
960 IF ASC(C1$)<>13 THEN 975
965 IF C$="" THEN 955
970 PRINT" ":RETURN
975 IF ASC(C1$)<>20 THEN 990
980 IF C$="" THEN 955
982 C1=LEN(C$):IF C1=1 THEN C$="":GOTO986
984 C$=LEFT$(C$,C1-1)
986 PRINT"■■ ■■";:GOTO955
990 PRINTC1$+"■■";:C$=C$+C1$:GOTO 955
1000 DATA"  ◠  "," ◡ "
1005 DATA"◤   ◥"
1010 DATA"■|  |■"
1015 DATA"■|  |■"
1020 DATA" ■|■ "
1025 DATA" ■|■ "
1030 DATA" ■|■ "
1035 DATA"‾‾__"
1040 DATA"    ","    ","
1045 DATA"—    ","  —  ","  — ","   —","\    ","  \  ","   \ ","|    ","  |  "
1046 DATA"  |  ","   |  ","|    ","|    ","|    ","|    ","|-   ","|    ","|    "
1050 DATA"|    "
1051 REM"                                  ■■■CATEGORIA 1■■
1055 DATA CPU,CHIP,RAM,SCORE,MEMORY,"DATA",TERMINAL,"FLOPPY DISC",CICLOS,LOOP
1060 DATA INCREMENTO,INPUT,LOGICO,OCTAL,SUBRUTINA,HEXADECIMAL,SUBSCRIPT
1065 DATA OUTPUT,PERIFERICO,BYTE
1066 REM"                                  ■■■CATEGORIA 2■■
1070 DATA BASEBALL,TENIS,BASKETBALL,FUTBOL,RUGBY,HOCKEY,SKIING,"WATER POLO"
1075 DATA VOLLEYBALL,BOXEO,"TIDDLEY WINKS",AJEDREZ,"BOCHAS  ",NATACION,KARATE
1079 REM"                                  ■■■CATEGORIA 3■■
1080 DATA "AUTOMOVILISMO",ESGRIMA,ARQUERIA,CICLISMO,"ALPINISMO"
1085 DATA PORSCHE,TOYOTA,VOLKSWAGEN,LINCOLN,PANTERA,LOTUS,TRIUMPH,CHEVROLET
1090 DATA DODGE,BMW,EDSEL,DATSUN,CONVERTIBLE,CADILLAC,CAPRI,"DUNE BUGGY"
1095 DATA MUSTANG,JALOPY,FERRARI,RAMBLER
1999 REM"                                  ■■■CATEGORIA 4■■
2000 DATA TAILANDIA,AMERICA,FRANCIA,ESPANIA,ALEMANIA,RUSIA,ITALIA,GRECIA
2010 DATA EGIPTO,RHODESIA,CHILE,CHINA,JAPAN,INDIA,CANADA,MEXICO,UGANDA
2020 DATA YUGOSLAVIA,GUAM,INGLATERRA
2024 REM"                                  ■■■CATEGORIA 5■■
2025 DATA PIZZA,TENSION,PESCADO,FRESCO,CIENTIFICO,LOCO,PIEDRA,ERROR
2030 DATA PELIGROSO,HUEVOS,CASTILLO,DISTINTO,LUGAR,ADIOS,ROPERO,LAGO,LANCHA
2035 DATA ERRONEAMENTE,EJEMPLO,PISTA
2500 END
```

GUIA PRACTICA GUIA PRACTICA

¿QUÉ ES EL CP/M?

Tal vez muchos no lo sepan, pero es un programa. Gracias a él la computadora tiene vida. Y, además, la posibilidad de correr una gran cantidad de software adicional.



CP/M significa Programa de Control para Microprocesadores y fue creado por la empresa norteamericana Digital Research.
En el nombre encontramos la primera pista: el CP/M es un programa. Pero no es un programa cualquiera, es un programa que da vida y permite el funcionamiento de una computadora dado que provee la administración de todos sus recursos, como ser el trabajo sobre la memoria, las operaciones de entrada y salida de datos entre la computadora y los periféricos (unidad de discos, impresora, modem, etcétera) y fundamentalmente el control sobre esos recursos.
Todo este cúmulo de palabras acaba de definir globalmente a un sistema operativo. Por tanto el CP/M es un sistema operativo. Una buena pregunta podría ser:
¿Por qué se hace hincapié en el CP/M? ¿Acaso la 64 o la 128 no tienen sistema operativo?
Por supuesto que la 128 y la 64 tienen sistema operativo, como dijimos antes, ninguna computadora podría funcionar sin él. Pero la incorporación del CP/M brinda la posibilidad de correr gran cantidad de programas adicionales con fines un poco más serios como archivos, controles de gestión, financieros, que ya se encuentran escritos para otras máquinas que trabajan con el sistema operativo CP/M. Y así resulta que nuestra 128 además de ser compatible con la 64, lo es también con aquellas computadoras con CP/M. ¿Interesante, no?
Pero hay una serie de particularidades que resultarán interesantes de mencionar.
El CP/M es un sistema operativo que fue diseñado para trabajar con el microprocesador 8080 de Intel o el Z-80 de Zilog. Esto explica la existencia del Z-80

junto al 8502 dentro de la 128.
Por otro lado el CP/M viene en un disco aparte, que debe estar colocado en la unidad de discos en el momento de encender la computadora para que se realice la carga del modo CP/M.
Esto se debe a que en el momento de encendido es el Z-80 quien toma el mando y sigue las instrucciones de un programa cargador que lo lleva a buscar ciertos datos en determinadas posiciones del disco colocado.
Si no las encuentra, el Z-80 se apaga y le transfiere el control al 8502 para que continúe con su rutina.
Si lo encuentra, se produce la carga del sistema operativo en la memoria RAM de la máquina y presenta en la pantalla el mensaje: A> con lo cual indica que el sistema está cargado y listo para recibir algún comando.
Pero atención, no se trata de comandos en un lenguaje determinado, sino propios del sistema operativo. En el caso de querer realizar algún programa en BASIC, o COBOL, o FORTRAN debemos cargar previamente el lenguaje deseado desde algún disco.
Todo esto nos puede resultar nuevo porque estamos acostumbrados a encender nuestra máquina y que se encuentre lista comprendiendo el comando Basic sin ningún paso previo.
Lo que ocurre es que tanto el modo 64 como el 128 vienen munidos con sus propios sistemas operativos y sus correspondientes BASIC en memorias ROM, de forma tal que el microprocesador encuentra toda la información necesaria para poner en funcionamiento al sistema sin necesidad del operador. Esto quiere decir que si tenemos un programa en Basic para CP/M y queremos correrlo debemos:
A) Cargar el CP/M
B) Cargar el intérprete Basic (que traduce cada instrucción Basic en instrucciones ejecutables por el Z-80).
C) Cargar el programa en cuestión.
Quizás suena un poco tedioso, pero ofrece como dijimos antes la posibilidad de un mayor repertorio de programas.

ESTRUCTURA DEL CP/M

El CP/M puede distinguirse en 4 subsistemas:
CCP (procesador de comandos de consola):
Trabaja como una interface entre el usuario que se expresa mediante el teclado y el CP/M, leyendo e interpretando los mensajes o comandos que llegan desde la consola.
BDOS (Sistema operativo básico de discos):
Conforma un gran número de instrucciones destinadas a supervisar el funcionamiento de las unidades de discos, procurando minimizar el tiempo de acceso a una determinada información almacenada en el disco.
TPA (Area de programas pasajeros):
Es la región de memoria donde se conservan los programas cargados desde el disco por los comandos del CCP anteriormente mencionados.
En la 128 esta área tiene una extensión de 59 kb. (El resto se lo lleva el CP/M).
BIOS (sistema básico de entrada y salida):
Provee las operaciones necesarias para acceder a las unidades de disco y los periféricos estandar.
Su constitución depende sustancialmente del hardware de la computadora y es quien posibilita la compatibilidad de los programas en distintas máquinas.
Esto es porque cuando el programa necesita acceder a una impresora, por ejemplo, no lo hace directamente sino que manda los datos al BIOS para que este los ubique.
Cada máquina tiene su propio BIOS y así, ante los requerimientos del programa se enviará la información al lugar correcto.
Por ello el fabricante de CP/M suministra la información necesaria del BIOS para que el usuario pueda adaptar sus periféricos a esta rutina.
Concluimos así esta nota esperando haber echado un poco más de luz sobre el tema CP/M.
Sólo queda por comentar que si bien el CP/M nos asegura compatibilidad entre distintas máquinas, nos corresponde a nosotros verificar que se trate del mismo CP/M, dado que existen distintas versiones y no todas compatibles.

Guillermo Fornaresio

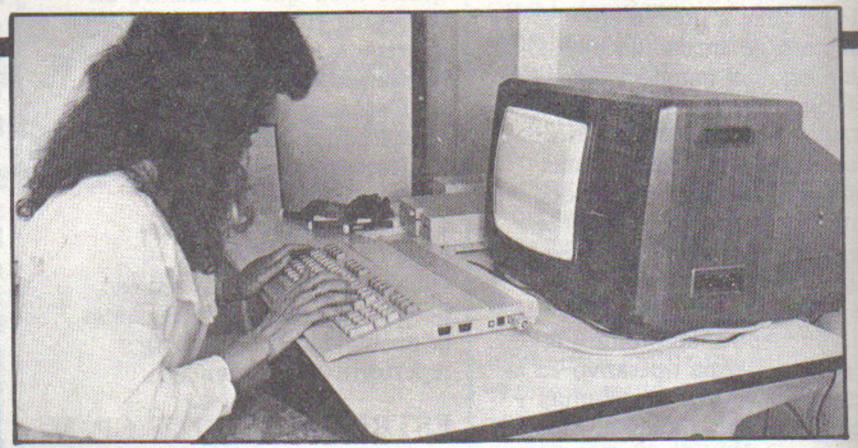
PARA SACARLE EL JUGO A LA MAQUINA

En el número anterior nos han hecho una nota, presentando al club de Usuarios oficial Drean Commodore.
En esa oportunidad contamos algunos servicios que tenemos, además de nuestro curso gratuito para nuevos usuarios, y cuáles eran nuestras filiales tanto en capital como en el Gran Bs. As. y el interior del país.
A partir de hoy y todos los meses en la revista va a estar esta página, "LA PAGINA DEL CLUB", o sea tu página.
En ella intentaremos brindarte novedades, y cosas que te sirvan para sacarle el jugo a tu Commodore.

Algunos datos útiles de la Drean Commodore 64

MEMORIA:

POKE 53272,21	mayúscula, gráficos
POKE 53272,23	minúscula, mayúscula
POKE 198,0	borra el buffer del teclado
POKE 808,225	desactiva stop, restore y list
POKE 808,237	reactiva stop, restore y list
POKE 53280,C	color borde pantalla
POKE 53281,C	color de pantalla
POKE 650,0	repite barra, cursor inst/del
POKE 650,96	no hay repetición de lo anterior
POKE 650,128	todas las teclas repiten
POKE 650,10	desactiva la repetición de las letras
POKE 774,0	proteger un programa
POKE 774,26	desproteger
SYS 64738	actua como apagar y encender (RESET)



COLORES:
0 NEGRO-1 BLANCO-2 ROJO
3 CYAN 4 PURPURA
5 VERDE 6 AZUL
7 AMARILLO 8 NARANJA
9 MARRON 10 ROJIZO
11 GRIS 1 12 GRIS 2
13 VERDOSO 14 CELESTE
15 GRIS 3

Servicios del Club

El club proporciona el curso básico gratuito en todas sus filiales. Para acceder al mismo, sólo es requisito presentarse en cualquiera de las filiales del club y reservar un horario. Por otra parte quienes así lo deseen, podrán asociarse al club.

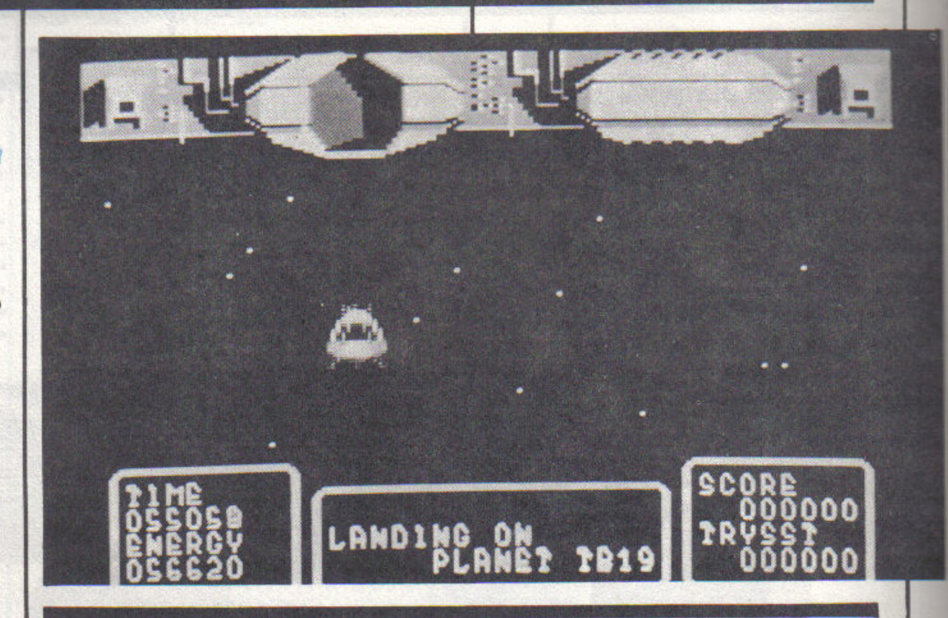
FILIALES DEL CLUB DREAN COMMODORE

LOCALIDAD	SEDE	DIRECCION	TELEFONO	C.P.
G BS AS	LOMAS	ACEVEDO 48	244-1257	1832
G BS AS	RAMOS	BART. MITRE 180	658-8665	1704
G BS AS	MARTINEZ	STA FE 1347	792-4985	1640
G BS AS	AVELLANEDA	MITRE 1755	203-5227	1870
G BS AS	SAN MARTIN	CALLE 52 3269	755-6559	1650
G BS AS	QUILMES	MORENO 609	253-6086	1878
BS AS	LA PLATA	CALLE 48 535 1P	021-249907	1900
BS AS	BAHIA B.	LAS HERAS 81/95	43201	8000
BS AS	M. PLATA	CATAMARCA 1755	43430	7600
CAPITAL	CENTRO	PUEYRREDON 860 9P	86-6430	1032
CAPITAL	CENTRO	RIVADAVIA 2450 4'A'	47-1805	1034
CAPITAL	BELGRANO	V. DE OBLIGADO 2833	70-6450	1429
CAPITAL	CABALLITO	J.B. ALBERDI 1196	431-1216	1406
CORDOBA	CORDOBA	JUJUY 574	33998	5000
CORDOBA	RIO CUARTO	VELEZ SARFIELD 62	21339	5800
MENDOZA	MENDOZA	INF. MERCEDES 78 2P	293790	5500
SALTA	SALTA	AV SARMIENTO 429	21-3920	4400
SANTA FE	SANTA FE	4 DE ENERO 2770	27445	3000
STA CRUZ	RIO GALLEG	SAN MARTIN 1201	8686	9400
TIERRA F.	USHUAIA	JAINEN 198	92156	9410
TUCUMAN	TUCUMAN	SAN JUAN 451	21-4331	4000
BS AS	TANDIL	RODRIGUEZ 769	22945	7000
CORRIENTES	CORRIENTES	JUNIN 1327 1° A		3400
E. RIOS	CONCORDIA	URQUIZA 742		3200

A: SUPERIOR.	D: REGULAR.
B: MUY BUENO.	E: MALO.
C: BUENO.	F: INACEPTABLE.

LANDING ON TB19

Rating Total: B
Creatividad: A
Profundidad del juego: B
Valor en relación al precio: Se justifica
Computadora: Drean Commodore 64
Editor:



Este es otro juego espacial, pero con la diferencia que aquí no hay que matar a nadie que solo debemos aterrizar nuestra nave en los distintos planetas del sistema solar.
Nuestra misión consiste en sacar el módulo de aterrizaje del "depósito" ubicado en la nave madre e ingresar dentro del planeta.
Claro que el acceso no es sencillo. El módulo puede ser gobernado desde el joystick.
A través de él podemos mover la nave hacia la derecha, hacia la izquierda y hacia arriba.
El acceso al primer planeta (TB19) es bastante difícil.
Primero debemos bajar suavemente hasta llegar a la superficie del planeta.
Cuando estemos a punto de ingresar en el interior del planeta deberemos controlar nuestra velocidad de descenso.
Una vez dentro, la siguiente pantalla en aparecer es el primer tramo de un túnel con paredes irregulares.
Este, por ser el primero, es el fácil. Todo derechito. Cuando nos acerquemos al final, ahí empieza el "baile".
El segundo tramo de túnel está "torcido" hacia la derecha. Baja diagonalmente y, por supuesto, sus paredes son superirregulares.
El tercer tramo del túnel, creemos que se lo pueden imaginar. Torcido pero para la izquierda. Es decir igualito que el anterior pero "pa' la izquierda".
El último tramo es cortito pero las paredes son mucho más irregulares. Si logramos esquivarlas sin chocar, lograremos aterrizar.
Una cálida musiquita anunciará el arribo, y veremos como nuestro puntaje aumenta.
"¿Y ahora a qué?" fue lo que nos preguntamos nosotros cuando logramos aterrizar. Algo que omitimos mencionar fue que en caso de "colisión" contra una de las paredes nuestra nave se desintegrará y deberemos comenzar desde arriba.
¿Se imaginan el chiste que nos causaba el hecho de chocar cuando estabamos a punto de aterrizar?
¡Les aseguramos que las carcajadas se escuchaban desde el obelisco!!
Una vez abajo tenemos que subir nuevamente a la nave nodriza, desde donde parte el módulo de aterrizaje.
Es decir que tenemos que esquivar nuevamente las peligrosas paredes del túnel y subir, que es más difícil que bajar.
Cuando descendíamos estabamos ayudados por la atracción de la gravedad, es decir que podíamos jugar con este parámetro y sólo acelerar para estabilizar la nave.
Pero cuando subimos, tenemos que aumentar la potencia de los motores lo justo, ni más ni menos. De lo contrario, a empezar de nuevo, pero desde abajo.
Si somos lo suficientemente hábiles como para salir del "pozo ciego", deberemos poner el módulo dentro de la nave madre.
Si lo logramos, escucharemos otra musiquita y nuestro puntaje se incrementará. A continuación se cerrarán las puertas del compartimiento de carga e iniciaremos un viaje hacia otro planeta.
El módulo se controla a través del joystick y del botón. Si movemos la palanca oprimiendo al mismo tiempo el botón reforzaremos los motores. Ideal para cuando estamos a punto de estrellarnos.
A LANDING ON TB19 le encontramos una interesante aplicación: practicar estacionamiento para así poner nuestro coche en lugares recontra reducidos. Como ven, no sólo se trata de un juego.

COLOR ME



Rating Total: B
Creatividad: B
Valor en relación al precio: Se justifica
Computadora: Drean Commodore 64
Editor: Mindscape

Este software fue diseñadó exclusivamente para que los niños de 6 a 89 años puedan divertirse realizando y pintando figuras que ellos crean.
Si disponemos de una impresora color o una impresora común, podremos imprimir nuestra creación en papel.
La primera pantalla en aparecer es la presentación (muy bien hecha). A continuación ingresamos en el modo edición de gráficos.
Aquí tenemos un "lápiz" que podemos moverlo a través del joystick o a través del Koala Pad (tableta gráfica).
Sobre la parte superior de la pantalla existe una línea de comandos con la cual podemos determinar algunas operaciones como:
Draw (dibujar)
Fill (pintar)
Menú (nos muestra todas las opciones)
Oops (restablece)
La primera de ella nos permite dibujar sobre la pantalla utilizando el joystick. Podemos utilizar distintos tipos de trazos (desde el grueso hasta el superfino).
Gracias a la selección del grosor del lápiz, podemos hacer cualquier tipo de gráficos.
Estos, como ya dijimos, los hacemos combinando el botón del joystick y el movimiento hacia uno de los ocho lados posibles.
Luego, si lo deseamos, podemos seleccionar la opción Fill (llenar) para colorear alguna sección del gráfico.
El colo puede seleccionarse de entre dieciseis posibles.
Con COLOR ME podemos suspender el proceso de coloreado presionando una sola vez el botón del joystick.
Cuando ingresamos a la opción Menú, nos muestra otras operaciones, las cuales pueden ser seleccionadas a través del joystick.
Aquí podemos optar por grabar nuestro diseño en disco, cargar otro gráfico desde el disco, borrar lo que esté en pantalla o cortar y pegar.
Esta última opción se refiere a lo siguiente: puede suceder que una parte de nuestro gráfico la usemos en otro lugar.
Para ello solo basta con marcar en el gráfico que parte queremos cortar y, luego, indicar donde la vamos a pegar.
Todo este traslado también se consigue accionando el joystick.
También desde la opción menú podemos regresar al Basic a través de Quit.
COLOR ME es ideal para los niños. De todas maneras siempre sucede lo que vemos en las propagandas: el pibe con cara de rabia y el padre jugando con la compu.

dBASE II

Rating Total: A
Creatividad: A
Valor en relación al precio: Se justifica
Computadora: Commodore 128 (modo CP/M)
Editor: Ashton Tate

A partir de este número comentaremos el excelente software disponible para la Commodore 128.
Comenzamos con dBASE II, que es una base de datos.
Para que se comprenda mejor, el dBASE II es un lenguaje de alto nivel orientado al manejo de datos. Es decir que tiene comandos y sentencias diseñadas exclusivamente para manipular todo tipo de información.
Detrás del dBASE II hay una pequeña historia. Esta comienza en Jet Propulsion Laboratory (JPL), empresa norteamericana dedicada a realizar cohetes y sondas espaciales.
En el año 1974 los científicos de la JPL comenzaron a utilizar programas avanzados de bases de datos para almacenar la información enviada por las sondas del JPL.
El programa se llamaba JPLDIS y su autor fue Jeb Lon científico de la empresa. El programa comenzó a utilizarse

```
. CREATE MAESTRO
ENTER RECORD STRUCTURE AS FOLLOWS:
 FIELD   NAME,TYPE,WIDTH,DECIMAL PLACES
 001     NOMBRE,C,50
 002     EDAD,N,2
 003     OCUPACION,C,70
 004     CARGO,C,40
 005
INPUT DATA NOW? N
. USE MAESTRO
. DISPLAY STRUCTURE
STRUCTURE FOR FILE:  MAESTRO.DBF
NUMBER OF RECORDS:   00000
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD      NAME        TYPE WIDTH   DEC
001      NOMBRE       C    050
002      EDAD         N    002
003      OCUPACION    C    070
004      CARGO        C    040
** TOTAL **               00163
```

Ratliff bautizó a su programa con el nombre de VULCAN y comenzó a promocionarlo con ese nombre.
Sus características no podían competir con los que se hallaban en el mercado. Así Ratliff creó una versión mejorada de VULCAN, el cual dejaba muy atrás a sus competidores.

En 1980 George Tate comenzó a oír las posibilidades de VULCAN. Tate era un distribuidor de software. Cuando probó el VULCAN quedó bastante sorprendido. Y más aún cuando supo que sólo se habían vendido 50 copias.

Pero Tate sorprendió más a Ratliff cuando le dijo que él podía vender 50 en un solo mes. De esta manera se firmó el contrato en donde se le otorgaba a Ratliff sustanciosos derechos de autor.

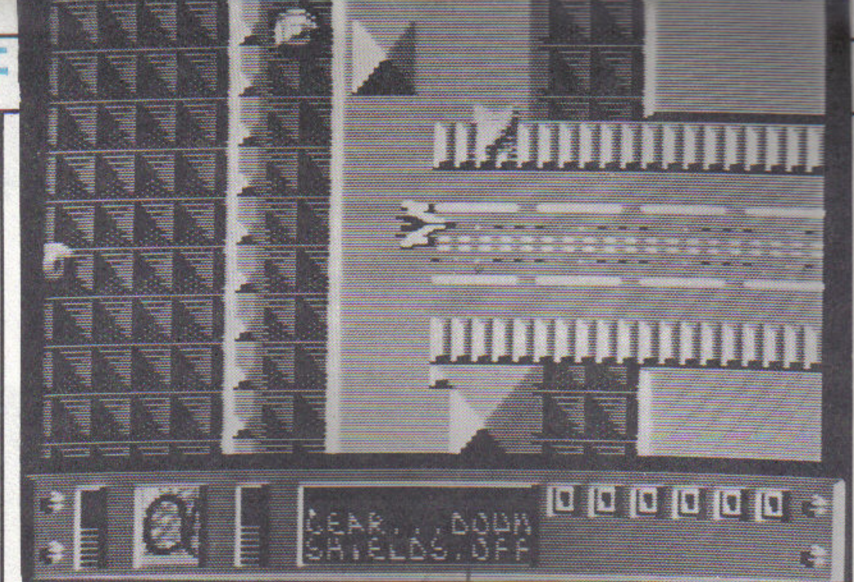
El nombre de VULCAN fue cambiado por dBASE II (nunca existió el dBASE I). Se inició una potente campaña publicitaria. Como resultado de ello Ashton-Tate comenzó a vender más de 2000 copias mensuales.
Luego de esta pequeña historia, analicemos un poco las ventajas y posibilidades de trabajar con el dBASE II. Admite trabajar con un máximo de 65535 registros. Cada uno de éstos puede tener hasta 32 campos, los cuales pueden ser numéricos, alfanuméricos o del tipo lógico. Gracias a la "pequeña" capacidad del dBASE II podemos manejar cualquier tipo de datos como ser stock, inventarios, etcétera. Lo primero que debemos hacer es definir nuestra base, es decir indicarle al programa la cantidad de campos que ella tendrá y el tipo.
Luego de definir la estructura del archivo podemos comenzar a ingresar uno a uno los registos que constituirán nuestra base. A partir de aquí podemos realizar distintas operaciones, como ser determinar cuantos registros cumplen con ciertas condiciones, totalizar los campos que cumplan ciertas especificaciones, etcétera.
Como en el Basic, podemos trabajar en modo directo o en modo programa. Es decir que podemos desarrollar programas específicos. Para ello el programador de dBASE dispone de sentencias orientadas a la programación estructurada.
Así, facilmente podemos confeccionar menúes, realizar reportes sobre impresora o video, añadir o eliminar registros, indexar los archivos, etcétera.
Como hemos mencionado, dBASE II trabaja bajo el control del CP/M. Para aquéllos que desconozcan este sistema operativo les sugerimos que lean en este mismo número una nota donde les explicamos qué es.

PARALLAX

Rating Total: B
Creatividad: A
Profundidad del juego: A
Valor en relación al precio: Se justifica
Computadora: Drean Commodore 64
Editor: Ocean

Tal vez tendríamos que calificar a este juego mirando, solamente, la presentación, que es sencillamente extraordinaria.
Nosotros deberemos manejar una nave de guerra que dispara cargas nucleares.
Saliendo de los tradicionales juegos de lucha espacial, PARALLAX simula una guerra en tres dimensiones. Es decir que tenemos en juego las tres coordenadas X, Y, Z.
Al inicio del combate se nos reporta el estado de la nave, del combustible y del oxígeno que tenemos.
Cuando terminamos la lectura del reporte, podemos tirar la palanca hacia adelante para tomar altura y comenzar a destruir las naves enemigas.
El combate transcurre dentro del fuerte espacial de un emperador, Pedro Enseguidavuelvo.
Este maligno y despiadado ser tratara de hacernos "pure".
El fuerte del emperador es bastante raro. Con la nave podemos ir en diferentes niveles por debajo o por arriba de los puestos de combate. En tanto, debemos esquivar los proyectiles que nos disparan. Por suerte nuestra nave puede temporalmente descender.
A través del botón del joystick pasamos a un menú de selección. Este sólo aparece cuando la nave ha aterrizado sobre algún sitio.
El menú nos permite salir al exterior. Para eso debemos suministrar suficiente oxígeno al piloto.
Una vez que estamos en el exterior, podemos iniciar nuestra caminata e ingresar dentro de alguna de las pirámides del emperador Enseguivuelvo.
Allí adentro encontraremos computadoras que nos pedirán nuestra clave de acceso. Si nos la reconocen podremos incrementar nuestro puntaje en varias o pocas unidades.
Para que el piloto pueda tomar más de una clave, aumentado así su posibilidad de acceso a la computadora, deberá matar a un secuaz del emperador, con lo cual tomará su clave.
No tenemos que descuidar que el oxígeno se consuma, por lo que debemos cumplir la visita a la pirámide rápidamente y luego volver a la nave.

La caminata hacia la nave se lleva a cabo con esa música parecida a la de las películas cuando un astronauta sale al espacio exterior.
Una vez dentro de la nave continuamos la lucha contra el malvado emperador, matando, si podemos, a sus fieles seguidores con sus naves "goma-goma". Ciertos proyectiles enemigos provocan que nuestros sistemas propulsores queden temporalmente suspendidos. Deberemos necesariamente esperar hasta que se restablezcan. Pero en ese lapso el enemigo continuará tirandonos

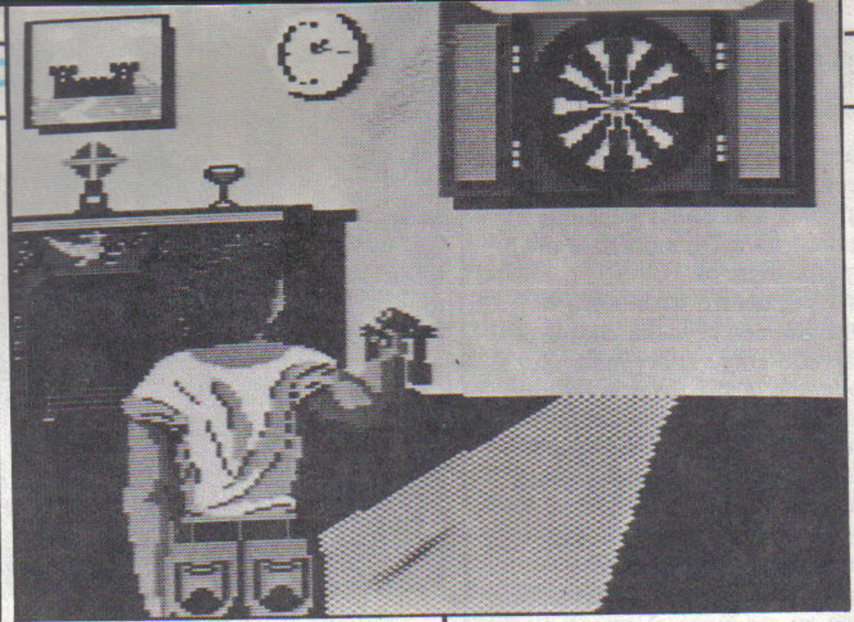


proyectiles. PARALLAX, con sus tres dimensiones, es un superjuego de combate. Los efectos visuales son "terribles".

INDOOR SPORTS

Rating Total: A
Creatividad: A
Profundidad del juego: B
Valor en relación al precio: Se justifica
Computadora: Drean Commodore 64
Editor: Mindscape

En el número anterior hemos comentado Ping Pong, un excelente juego de tenis de mesa. Los mismos editores han desarrollado INDOOR SPORTS, con el que podemos practicar bowling, dardos y tejo.
Pueden participar hasta cuatro jugadores. Al comienzo se nos pide el nombre de cada participante.
Si sólo queremos jugar contra la computadora debemos ingresar nuestro nombre y oprimir la tecla RETURN por cada pregunta.
A partir de aquí podemos seleccionar el juego deseado. Es decir podemos decirle a la computadora si vamos a jugar al bowling, al tejo, o si vamos a practicar tiro al blanco con dardos.
Si optamos por este último juego, indicaremos al INDOOR SPORTS las condiciones de la competencia. Es decir debemos decirle si vamos a jugar contra la computadora, el total de puntaje obtenido para ganar, etcétera.
En esta competencia gana el primero que llega a cero. Si nosotros, por ejemplo, comenzamos desde 301 puntos, por cada dardo bien puesto (es decir obtener un buen acierto) irá disminuyendo el puntaje parcial hasta llegar a cero.
La pantalla que sigue a continuación nos permite tomar el dardo y establecer el ángulo de disparo y la fuerza con que se disparará.
El otro juego es el Tejo. Para aquellos que no lo conozcan, se desarrolla sobre una mesa con pequeños orificios por donde sale aire con el fin de darle velocidad al disco de plástico que se desliza sin rozamientos, participan dos personas y deben tratar de golpear al disco de tal manera que éste ingrese en el marco del contrario.
El que nos muestra INDOOR SPORTS es totalmente idéntico



al verdadero. Es más, cuando comienza se escucha cómo se prende el compresor neumático, suministrando el aire necesario para provocar el mínimo rozamiento con la superficie.
Es posible, como con los dardos, indicar las condiciones de juego, como la velocidad de la ficha, si vamos a cambiar de lado después de cada partido o si jugamos o no contra la computadora.
Si queremos "gozar" viendo como juega la computadora sola (opción demostración), ponemos la velocidad del juego en Hyperdrive. Es aconsejable, alejarse de la pantalla por si se sale la ficha de la pantalla.
El tercer y último juego es el bowling. Aquí tenemos que, como en el bowling original, "acabar" con la mayor cantidad de palos.
El juego cumple cien por cien con las condiciones de esta competencia. Es decir que podemos darle efecto a la bola, tirar con estilo o, si somos "perros", hacer una supercanaleta.
INDOOR SPORTS es, realmente, un excelente juego.
Claro que, como siempre, tenemos que dar nuestro útil consejo: Tengan cuidado en el lanzamiento de dardos. El cadete de la editorial tuvo que ser llevado con urgencia al hospital debido a que un inconciente apuntó para cualquier lado.

SUPERCYCLE

Rating Total: n
Creatividad: n
Profundidad del juego: n
Valor en relación al precio: Se justifica
Computadora: Dream Commodore 64
Editor: Epyx

A partir de SUPERCYCLE se ha roto el "mito" por el que todas las competencias sobre pistas eran exclusivamente para coches.
Epyx, la gran empresa norteamericana de software, lanzó el SUPERCYCLE, una carrera de motos a todo "trapo".
Nuestro piloto, Mike Agarramequemevuelo, tendrá que manejar hábilmente la moto. A través del Joystick le diremos a Mike para qué lado deberá moverse.
El juego comienza preguntándonos sobre las características de nuestro equipo de competición, es decir el color y diseño de nuestro buzo antiflama y el color de la moto.
Luego tendremos que seleccionar el nivel de competencia que puede ser 1, 2 o 3, siendo el primero el más fácil y el último el más difícil.
La siguiente pantalla nos lleva a la pista de competición. José Bandera se encuentra a un costado esperando que presionemos el botón del joystick para darnos la señal de partida.
Mike se ubica en el medio de los dos competidores que largan (son tres en total). Para que pueda tener control pleno de la máquina, Mike dispone de relojes.
Estos son el velocímetro, el cuenta revoluciones y el indicador de cambios (sólo tiene tres). Estos últimos se accionan a través del botón del joystick.
Cuando José Bandera da la señal, tenemos que tirar la palanca hacia adelante y controlar el cuenta revoluciones.
Cuando se aproxime a la parte roja, "tiramos" el segundo cambio y repetimos la operación hasta meter la tercera.
Para frenar podemos optar por dos formas: tirar la palanca hacia atrás o meter un rebaje.
De aquí en más debemos cumplir la vuelta en un determinado tiempo. Para ello tenemos que cuidar de no irnos de pista o de no chocar reiteradamente a los demás competidores.
En caso de cumplir con la vuelta en el tiempo prefijado, pasamos a competir en otro circuito.
Hay dos circuitos internacionales. En el primero, el paisaje es un valle montañoso, y hay una pista "suave" sin demasiadas curvas.
En la segunda nos encontramos en el desierto norteamericano. Aquí tenemos que esquivar cactus si no queremos llenarnos de pinches molestos.
Ya en la tercera, entramos en la ciudad. El cuarto circuito está en las afueras de la ciudad. Aquí podremos ir incrementando nuestro puntaje a través de banderines que se encuentran en el piso. Basta con pasar sobre ellos para lograr un puntaje mucho mayor. Algo que olvidamos mencionar es que si Mike choca puede retomar la carrera. Sólo se acaba la competencia si él no cumplió con la vuelta en el tiempo fijado.

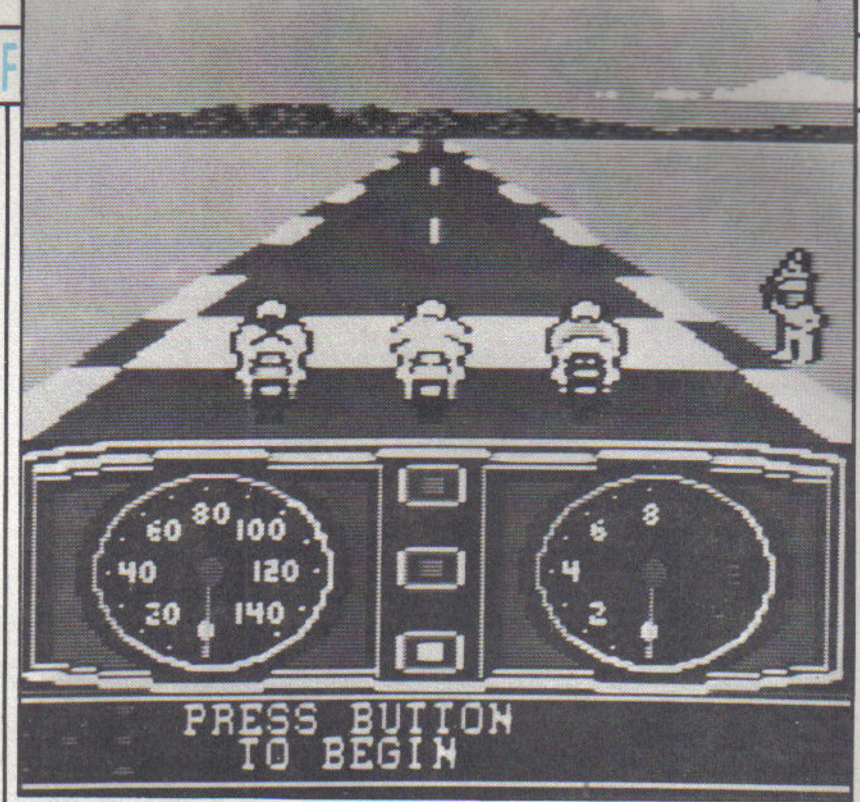
La pista correspondiente a la quinta carrera se encuentra atravesando el mar. Ya comienzan a presentarse los obstáculos: baches que podríamos llamar "pozos ciegos".

La sexta competencia se realiza bajo un frío intenso. Mientras corremos podemos ver el hielo que hay a los costados.

La séptima transcurre de noche habiendo manchas de aceite peligrosísimas.

De esta manera cada competencia tiene sus propios escenarios y clima de acción (en una de las carreras hay una tormenta terrible).

Luego de cumplir con el tiempo previsto para la décimo segunda carrera (este tiempo disminuye de carrera en carrera), aparece nuestra posición general y la posibilidad de grabar nuestro nombre junto con los demás records.



Algo importante: hay ocasiones en que, si vamos muy rápido, ¡Mike se quiera bajar de la moto!
¿Por qué le habrán puesto Mike Agarramequemevuelo?

DIRECTORIO

Estoy realmente satisfecho con las secciones tratadas en la revista y es por ello que ahora quiero integrarme a ustedes haciéndoles dos consultas:
1) En Trucos del número 4 (pág. 29) hay una nota titulada El buffer del teclado; explica la nota la carga de un programa en cinta pero sin el uso de comandos, es decir lo hacen por programa. Yo necesito saber cómo puedo hacer lo mismo pero para cargar programas del disco (en especial el directorio). Hice un intento con Trucos del número 2 (pág. 33) en una nota que dice directorio inofensivo pero está en modo directo y yo lo quiero hacer en un programa BASIC.
2) Esta consulta está vinculada con la primera, es decir, como podría hacer para cargar el directorio pero en Assembler. Estuve leyendo y creo que lo que tengo que hacer es usar las rutinas SETFLS y SETNAM y luego la rutina LOAD. El problema lo tengo con la forma en que debo trabajar con las rutina SETFLS y SETNAM.
SETFLS me dice que cargue en A el número del fichero lógico y en X el número de dispositivo (para discos es 8/9) ¿Qué número cargo en A? ¿cualquiera entre 2 y 14?; luego debo llamar la rutina.
Después debo usar SETNAM que me dice que cargue A con la longitud del nombre del fichero.
Si es un fichero cualquiera la longitud es la cantidad de caracteres, pero ¿cuál es la dirección del nombre del fichero?
Y luego dice que si no se requiere nombre, cargue A con ceros; ¿es el caso del directorio? ¿Cuál es el nombre del directorio? ¿es $? ¿y cuál es la dirección del nombre del directorio?
Ultima: ¿debo usar sólo estas tres rutinas o algunas más?
Mis felicitaciones. Muchas gracias.
Alejandro R. Coloma. Quilmes.

1) Si utilizas el buffer del teclado para ejecutar comandos en modo programa, el directorio lo puedes cargar realizando las siguientes instrucciones:

```
10 POKE631,76:POKE632,207:
POKE633,34:POKE634,36:POK-
E635,34:POKE636,44
20 POKE637,56:POKE638,13:
POKE198,8
```

Estas sentencias las puedes poner tanto en modo directo como en modo programa.
2) Desde el Assembler puedes tomarlo a través de:

```
LDA  $08
TAX
LDY  $FF
JSR $FFBA
LDA  $01
LDX  $00
LDY  $C1
JSR $FFBD
LDA  $00
LDX  $00
LDY  $04
JSR $FF05
RTS
```

Además, en la dirección $C100, tendrás que poner el valor 24 (hexadecimal) correspondiente al código ASCII de "$".
SETLFS pide que en el acumulador pongas el número de archivo, cualquiera entre 1 y 255. Te conviene, si vas a trabajar con disco, poner 8.
En X se debe poner el número de dispositivo (8 o 9).
En el registro Y va la dirección secundaria, es decir el comando. En este caso nosotros pedimos, a través de $FF, que se cargue a partir de la dirección que se indica en la rutina LOAD.
Luego llamamos a la rutina SETNAM. Aquí se debe poner la cantidad de caracteres del nombre del archivo en el acumulador y la dirección en donde se encuentra, en los registros X e Y, en formato parte baja-alta.
Por ello es que en la dirección $C100 deberas poner el ASCII de "$".

UTILIPRINT

Ante todo quiero manifestarles mi agrado al notar un permanente afán de mejora de su publicación de la que soy permanente lector. En el Nro. 9 me entusiasmó el programa UTILIPRINT que entiendo sería de gran utilidad —SI FUNCIONARA—.
Soy un principiante y del código máquina entiendo un pepino, de manera que lo tipié y controlé muy cuidadosamente.
Al correr el programa éste se detiene en la 100 y aparece OUT OF DATA y es lógico que así sea pues al programa le sobra un READ o le falta un DATA. En principio asumí lo primero y limité I a 49648 pero me dio ERROR EN DATAS; imprimí la sumatoria y resultó X = 55783 en lugar de ..84. Aquí ya se me quemaron los papeles y se terminó mi sapienza... por favor !!HELP!!

Continuamos con esta sección para que los lectores planteen sus consultas y sugerencias. Para eso deben escribir a Revista para usuarios de Drean Commodore, Paraná 720, 5to. Piso, (1017) Cap.

Algo más... Quisiera que explicaran más detenidamente su utilización práctica. No comprendo cómo debo hacer para enganchar el UTILIPRINT que tengo como una subrutina en disco, en una o varias partes de un mismo program.
Tengo idea de que la mayoría de los lectores de su revista así como de otras similares son por lo general principiantes y lo que para los autores es trivial u obvio no lo es tanto para nosotros.
Les saluda muy cordialmente.
Jorge L. S. Spiridonidis
Capital Federal

Debemos decirte estimado Jorge que encontramos un par de errores en el listado que nos enviastes.
Uno fue culpa de nuestra impresora ya que no se entiende si es 3 u 8. En el otro lado hubo un error tuyo.
Toma nota: el primer valor de la línea 150 es 169 y no 163.
El octavo valor de la línea 410 debe ser 193 y no 198.
Todo lo otro está bien, por lo que no entendemos por que te da OUT OF DATA. Por favor verificalo de nuevo.
UTILIPRINT nos permite diseñar nuestras propia pantallas usando para ello el editor de pantalla que tiene la Drean Commodore 64.
Para que lo entiendas mejor imaginate lo siguiente: queres poner un mensaje del tipo "OPRIMA LA LETRA A PARA TAL TAREA O LA LETRA B PARA OTRO O F PARA FINALIZAR".
Este no entra todo en la pantalla, por lo que debe ir probando y corrigiendo sobre las sentencias PRINT que permiten visualizarlo.
Si usas UTILIPRINT podes escribir en pantalla directamente el mensaje anterior e ir poniendo el texto como vos queres y una sola vez.
Luego te vas con el cursor a la primera línea y allí, sólo allí escribís SYS49152.
UTILIPRINT te pasa lo que tenes en pantalla a tu Programa Basic poniéndolo al final y usando el incremento entre línea y línea que vos le hayas dado.
Con SYS49152 el incremento es de 10. Si, en cambio, tipeas SYS49152,90 el incremento de cada línea será de 90.



LOS CAZAFANTASMAS
MATEMATICAS
TENIS INTERNACIONAL
POPEYE
commodore
MPS 801
Futbol Internacional de Commodore
LOGO
UN LENGUAJE PARA APRENDER
Software/Hardware
PEEK
para...
Drean commodore